

A machine learning-based surrogate model to optimize
performance of large air-to-water heat-pump in existing
buildings

Lucile Favero Montero

Supervisor: Sylvain Sardy

Collaborator: Omar Montero Dominguez

Department of Mathematics, faculty of Science
University of Geneva

September 2022

Abstract

Buildings in Switzerland account for over 50% of total energy consumption and 24% of CO_2 emissions, making them one of the most significant sectors for substantial decarbonization*. Parallel to decreasing building stock heat demand, a huge shift from fossil to renewable energy, particularly via heat-pumps* systems, may dramatically cut building stock CO_2 emissions. Thus numerical simulation models of the heat-pump system were constructed and validated based on a monitoring campaign in a pilot project. The simulation enables us to alter particular system input parameters and observe how doing so affects the performance and CO_2 emissions. One goal can be to test a variety of input values in an attempt to optimize the system. However, due to the computational time required for each simulation, this approach is not feasible, especially for multi-objective optimization in a large input space, where a lot of simulations are frequently needed.

In this study, we propose a surrogate machine learning model that mimics the numerical simulation while cutting down on calculation time. The aim is to find the range of input parameters that maximize the performance of the heat-pump while reducing CO_2 emissions.

Keywords: Air-to-water heat-pump, existing buildings, TRNSYS simulation, system performance, CO_2 emissions, surrogate model, machine learning, Extreme Gradient Boosting

Table of Contents

Abstract	i
List of Acronyms	vi
List of Figures	x
List of Tables	xi
1 Introduction	1
1.1 Background and related work	1
1.2 The approach	2
1.3 Thesis Outline	3
1.4 Heating system simulation	4
2 Machine learning	7
2.1 Framework	7
2.1.1 Machine learning definition	7
2.1.2 Regression framework	8
2.1.3 Goodness of predictive model	8
2.1.4 Hyperparameter and cross-validation	9
2.1.5 Bias and variance	9
2.1.6 Model evaluation	10
2.2 Model implementation	11
2.2.1 ML workflow	11
2.2.2 Models used in this study	13

3 Model implementation	14
3.1 The dataset	14
3.1.1 Generation of the dataset	14
3.1.2 Inputs distribution	16
3.1.3 Outputs distribution	16
3.1.4 Correlation between outputs	17
3.2 Model implementation	18
3.2.1 Linear model	18
3.2.2 More complex models	20
3.3 Model comparison	30
3.3.1 COP	31
3.3.2 CO_2	32
3.3.3 Summary and model selection	33
4 Final model and optimization	34
4.1 Final model implementation	34
4.1.1 Features selections	34
4.1.2 Reduced XGB implementation and analysis	36
4.1.3 Comparison with other reduced models	40
4.2 Predictive space	41
4.3 Optimization on reduced space	45
4.3.1 Distribution of predictors in optimal regions	45
4.3.2 Results	47
4.3.3 Verification	47
4.3.4 Discussion	49
5 Conclusion	50
6 Appendix	52

6.1	Learning curves of linear model	52
6.2	Model hyperparameter selection	53
6.3	Comparison of the models with MSE	54
6.4	Glossary	55

List of Acronyms

The meaning of each acronym and abbreviation used in the thesis is explained below. The page where each one is declared or used for the first time is also included.

ASHP air-source heat pump	1
CART Classification And Regression Trees	10
COP Coefficient of performance	3
DHW Domestic hot water	6
HP Heat Pump	1
KNN K-nearest neighbors	10
LHS Latin Hypercube Sampling	15
ML Machine-learning	3

MSE Mean Squared Error	9
R² R-squared	9
RF Random Forest	10
SH Space Heating	6
SVR Support Vector Regression	10
XGB Extreme Gradient Boosting	10

List of Figures

1.1	Simplified diagram of the simulated heating system	4
1.2	TRNSYS simulation inputs and outputs	5
2.1	Machine learning workflow: steps to implement a given model and compare it with the others.	12
3.1	Workflow to create the dataset	14
3.2	Input normalized distributions for each predictor. Each grey dot is an input. Black dots are the normalized default values.	16
3.3	Outputs distribution: COP and CO_2 emissions.	17
3.4	Correlation matrix of the simulation outputs	17
3.5	Distribution of the COP against the CO_2 emissions	18
3.6	Prediction error of the linear model	19
3.7	Residual distribution of the linear model	19
3.8	Outlier detection of the linear model with Cook's distance	20
3.9	Prediction error for the SVR	21
3.10	Prediction error for the KNN	22
3.11	Prediction error for the CART	22
3.12	Prediction error for the RF	23
3.13	Prediction error for the XGB	23
3.14	Residual distribution for the SVR	24
3.15	Residual distribution for the KNN	25

3.16	Residual distribution for the CART	25
3.17	Residual distribution for the RF	26
3.18	Residual distribution for the XGB	26
3.19	Learning curves for the SVR	27
3.20	Learning curves for the KNN	28
3.21	Learning curves for the CART	28
3.22	Learning curves for the RF	29
3.23	Learning curves for the XGB	29
3.24	Model comparison with violin plot of the COP R^2 on train set. 5 folds CV, 3 repetitions.	31
3.25	Model comparison with violin plot of the COP R^2 on test set. 5 folds CV, 3 repetitions.	31
3.26	Model comparison with violin plot of the $CO_2 R^2$ on train set. 5 folds CV, 3 repetitions.	32
3.27	Model comparison with violin plot of the $CO_2 R^2$ on test set. 5 folds CV, 3 repetitions.	33
4.1	Recursive feature elimination for XGB	35
4.2	Features ranked by the explained variance that each feature adds to the model	35
4.3	Final model implementation workflow	36
4.4	Learning rate selection for the final model	37
4.5	Maximum depth selection for the final model	37
4.6	Number of estimators selection for the final model	38
4.7	Subsample selection for the final model	38
4.8	Prediction error for the final model	39
4.9	Residual distribution for the final model	39
4.10	Learning curves for the final model	40
4.11	Reduced model comparison with violin plot of the COP R^2 on train set. 5 folds CV, 3 repetitions.	41

4.12 Reduced model comparison with violin plot of the COP R^2 on test set. 5 folds CV, 3 repetitions.	41
4.13 Reduced model comparison with violin plot of the CO_2 R^2 on train set. 5 folds CV, 3 repetitions.	42
4.14 Reduced model comparison with violin plot of the CO_2 R^2 on test set. 5 folds CV, 3 repetitions.	42
4.15 Slice of the predictive space displaying the SH-HP heating curve and DHW demand predictors	43
4.16 Slice of the predictive space displaying the SH tank heating curve and SH-HP heating curve predictors	44
4.17 Slice of the predictive space displaying SH-HP heating curve and SH volume tank predictors	44
4.18 Violin plot of the normalized distribution of the predictors that provide $y_{COP} > 2.80$. Default values are signaled with a black dot.	45
4.19 Violin plot of the normalized distribution of the predictors that provide $y_{CO_2} < 23.66$. Default values are signaled with a black dot.	46
4.20 Violin plot of the normalized distribution of the predictors that provide $y_{COP} > 2.80$ and $y_{CO_2} < 23.66$. Default values are signaled with a black dot.	46
4.21 Input normalized distributions for each variable of the verification data set. Each grey dot is an input. Black dots are the normalized default values.	48
4.22 Distribution of the outputs for data sampled in the default space vs data sampled in the optimal space	48
4.23 Changeability of the features	49
6.1 Learning curves of the linear model	52
6.2 Total budget selection for SVR	53
6.3 Number of neighbors selection for KNN	53
6.4 Number of estimators selection for RF	54
6.5 Maximum depth selection for RF	54

6.6	Learning rate selection for XGB	55
6.7	Maximum depth selection for XGB	55
6.8	Number of estimators selection for XGB	56
6.9	Subsample selection for XGB	56
6.10	Model comparison with violin plot of the COP MSE on train set. 5 folds CV, 3 repetitions.	56
6.11	Model comparison with violin plot of the COP MSE on test set. 5 folds CV, 3 repetitions.	57
6.12	Model comparison with violin plot of the CO_2 MSE on train set. 5 folds CV, 3 repetitions.	57
6.13	Model comparison with violin plot of the CO_2 MSE on test set. 5 folds CV, 3 repetitions.	57

List of Tables

1.1	Heat-pump simulation's parameters	6
2.1	Model definition and characteristics	13
3.1	Range of input predictors	15
3.2	Results of the predicted error plots	24
3.3	Results of residual distribution plots	27
3.4	Results of COP learning curves plots.	30
3.5	Results CO_2 emissions learning curves plots.	30
3.6	Summary of the model analysis	33
4.1	Comparison of the default space and optimal space for the COP and CO_2 . .	48
4.2	Different levels of changeability of the predictors	49
6.1	Glossary related to technical heat-pump and ML terms.	58

Chapter 1

Introduction

1.1 Background and related work

In Switzerland, buildings account for nearly 50 % (100 TWh/yr) of the final energy demand and 24 % (11.2 Mio. t/yr) of the CO₂ emissions OFEV. (2022a), OFEV. (2022b), making them one of the most significant sectors for substantial decarbonization *. Parallel to decreasing building stock heat demand, a huge shift from fossil to renewable energy, particularly via Heat Pump (HP) * systems, may drastically cut building stock CO₂ emissions Montero et al. (2020).

In dense urban areas, air-to-water air-source heat pump (ASHP) technology often turns out to be the only available option for replacing fossil fuel boilers, since other renewable energy sources are often limited: too long distance to a lake or river, lack of boreholes, limited solar energy due to roof size, etc. De Sousa Fraga (2017). In such a situation, the only widely accessible and abundant renewable energy source for heat pumps is outside air.

While the potential HP Swiss market is huge, many challenges and obstacles need to be solved, in particular for the large air-to-water ASHPs in MFBs. One of the most significant technical challenges for the implementation of air-to-water ASHP in MFBs is the lack of commercially available machines with a heating capacity above 50 kWth, specifically

designed for the residential sector [Montero et al. \(2022\)](#). The lack of large HP models can be attributed to the complexity of their implementation in MFBs .

Within this context, the local public utility Services Industriels de Genève is developing a series of pilot projects concerning the replacement of fossil-based heating systems by large air-to water ASHP in MFB. In collaboration with the university of Geneva, these projects were monitored (for two years) and analyzed to study the performance and technical constraints in real conditions of use. Simultaneously, numerical simulation models of the system were created in TRNSYS* and validated based on on-site measurements of the pilot project [Montero et al. \(2022\)](#).

The simulation enables us to alter particular system input parameters and observe how doing so affects the performance and CO_2 emitted. One goal can be to test a variety of input values in an attempt to optimize the system i.e. maximize the performance of the heat-pump while reducing CO_2 emissions.

However, due to the computational time required for each simulation, this approach is not feasible, especially for multi-objective optimization in a large input space, where a lot of simulations are frequently needed. The challenge here is to find a no-time-consuming model that mimics the simulation with good accuracy and use it to run the optimization.

1.2 The approach

Buildings are nonlinear systems that exhibit complicated and dynamic behaviour. They have a sizable number of parts and systems that have a substantial impact on how energy-efficient they are. Due to this complexity, it is challenging to optimize the energy performance of the entire structure while taking energy consumption and environmental consequences into account. [Sharif and Hammad \(2019\)](#)

The energy performance of buildings has been visualized, analyzed, optimized, and predicted using a variety of mathematical, statistical, and computer models. These strategies for enhancing building energy performance range widely, from elementary mathematics

to the most intricate neural networks Abdallah (2014). Models that mimic computationally expensive building simulation models are called **surrogate models***. Nowadays, there exists a large diversity of surrogate models in the field, where some of them are Machine-learning (**ML**)* based (e.g. Seyedzadeh et al. (2020),Kalogirou and Bojic (2000), Dallapiccola et al. (2020),Magnier and Haghigat (2010),Sharif and Hammad (2019), Sharif and Hammad (2019) or Thrampoulidis et al. (2021)).

In this study, we propose a surrogate machine learning model that mimics the numerical simulation with good accuracy while cutting down on calculation time. The aim is to find the range of input parameters that maximize the performance of the heat-pump while reducing CO_2 emissions. Our approach consists of selecting a model that uses only the most important parameters while being capable of good predictions. Thus it allows finding the predictor range that maximizes Coefficient of performance (**COP**) and minimizes CO_2 by spanning this reduced predicted space.

1.3 Thesis Outline

The thesis is structured as follows:

- We start by outlining the fundamentals of machine learning. We address the subjects of regression framework, goodness of the predictive model, hyperparameter selection, via cross-validation, bias and variance trade-off, and model evaluation. The models used in this thesis are briefly defined after that.
- Then these models are implemented, analyzed, and compared to select the best one.
- After selecting the important features, we train and analyze the final model on this reduced input space. The model set, we can now perform predictions over the whole reduced space, and detect regions that lead to high COP and low CO_2 emissions.

1.4 Heating system simulation

The numerical heating system simulation is implemented in TRNSYS Klein et al. (2017).

This software is designed to address complex energy system problems, by solving differential equations. Figure 1.1 shows a simplified diagram of the simulated heating system, and Figure 1.2 depicts simulation inputs and outputs. The inputs are hydraulic components and regulation parameters of the heating system. Table 1.1 provides definitions for each of them. The first output is the coefficient of performance (COP), computed as follows:

$$COP = \frac{Q_{prod}}{E_{elec}}$$

where Q_{prod} is the heat produced by the heat-pump and E_{elec} is the electricity consumed.

The second output is the CO_2 emission.

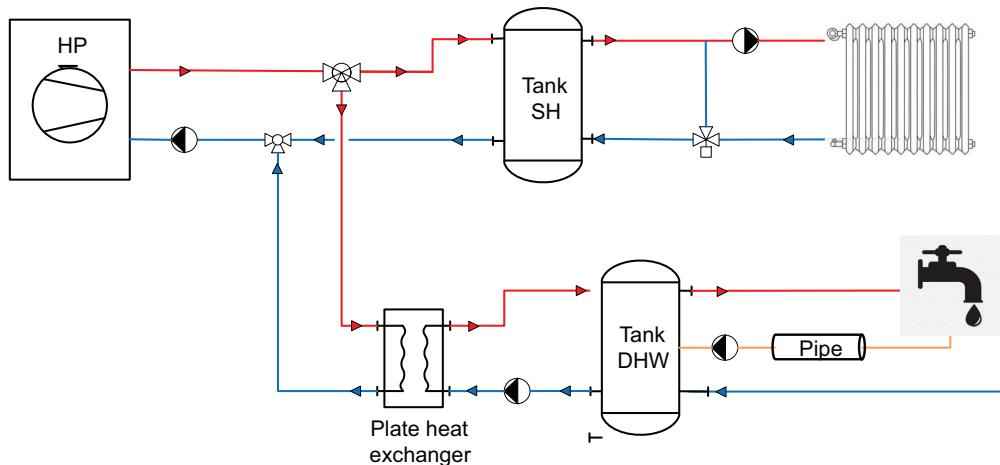


Figure 1.1: Simplified diagram of the simulated heating system

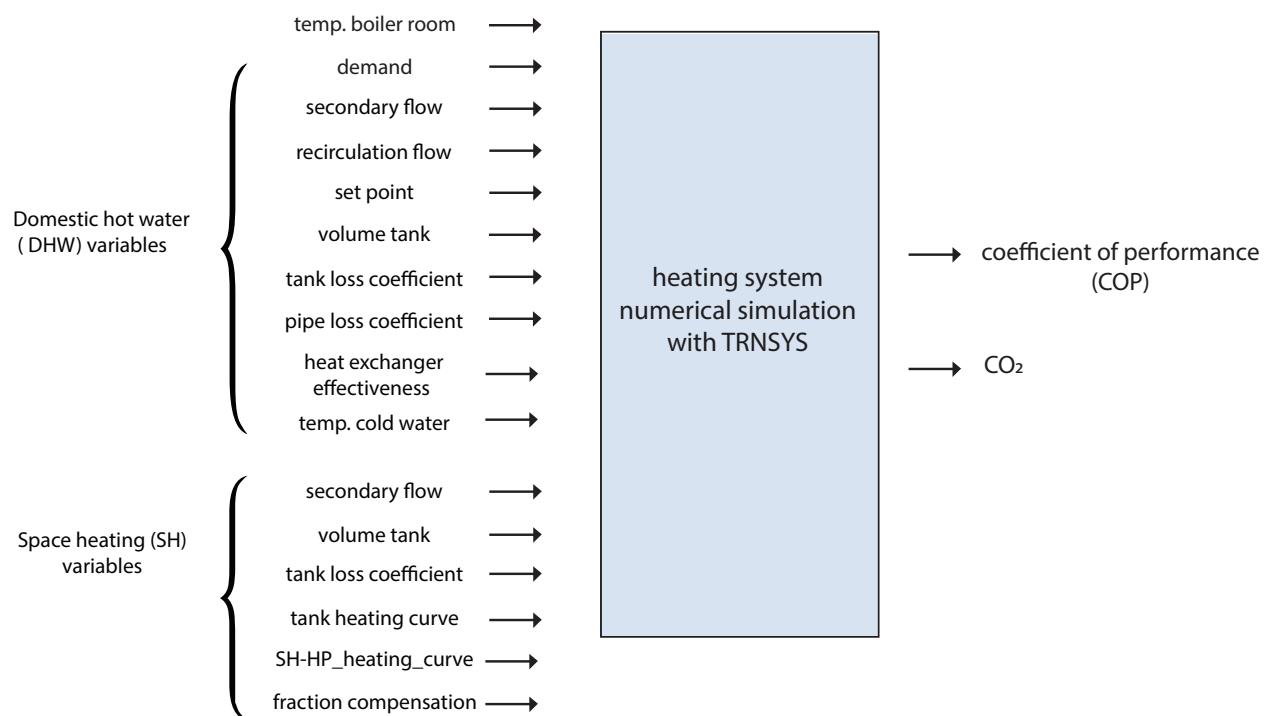


Figure 1.2: TRNSYS simulation inputs and outputs

Table 1.1: Heat-pump simulation's parameters

Parameter	Description	Unit
DHW demand	Domestic hot water (DHW) demand as the flow consumed by the building's tenants	<i>L/day</i>
DHW secondary flow	Circulator flow rate to fill the DHW tank	<i>m³/hour</i>
DHW recirculation flow	Waterflow of the DHW recirculation loop	<i>m³/hour</i>
Temperature cold water	Temperature of the sanitary water in the city's distribution network	°C
DHW volume tank	Volume for the storage of the production of the heat pump cycles for DHW	<i>m³</i>
SH volume tank	Volume for the storage of the production of the HP cycles for Space Heating (SH)	<i>m³</i>
SH secondary flow	Distribution flow to the radiators	<i>m³/hour</i>
SH fraction compensation	Energy fraction of space heating compensation	%
Temperature boiler room	Temperature of the technical room where the tanks, heat exchangers, distribution systems, etc. are located	°C
Tank loss coefficient	Heat loss coefficient of the tanks corresponding to the thickness of the mineral wool insulation	<i>W/m²K</i>
DHW pipe loss coefficient	Heat loss coefficient of the pipe insulation which corresponds to the thickness of the mineral wool insulation	<i>W/m²K</i>
SH-HP heating curve	Increase in the temperature of the HP in relation to the heating distribution system	°C
SH tank heating curve	Relationship between the heating system supply temperature for space heating and the outside air temperature	°C
DHW heat exchanger effectiveness	Heat exchanger effectiveness	no unit
DHW set point	DHW set point temperature	°C

Chapter 2

Machine learning

This part is based on the ML reference books: [James et al. \(2013\)](#) and [Hastie et al. \(2009\)](#).

2.1 Framework

2.1.1 Machine learning definition

Machine learning aims at programming a computer to learn to perform a task (prediction, inference ...) from information (data) in an optimal way in terms of performance measures. Learning types fall into two main categories: supervised and unsupervised learning.

During **supervised learning** we observe n training data $(x_i, y_i) i \in \{1, \dots, n\}$ where the inputs $x_i \in \mathbb{R}^p$ are the predictors or features, and the outputs y_i are the responses. The goal is to predict the response y_0 for a new predictor x_0 . We have a **regression** problem when the responses are quantitative, and a **classification** problem when they are qualitative.

During **unsupervised learning**, we only observe multivariate data $x_i \in \mathbb{R}^p i \in \{1, \dots, n\}$ without a particular response. The aim is to understand, interpret and summarize the data.

In this study, we are in the case of regression supervised learning.

2.1.2 Regression framework

In a stochastic framework, we have a quantitative response Y and p predictors X_1, \dots, X_p . We assume the following relationship between them:

$$Y = f(X) + \epsilon$$

where ϵ is a random error term with $E(\epsilon) = 0$ and $\text{var}(\epsilon) = \sigma^2$. The function f represents the systematic information that X provides to Y . It is fixed and unknown.

In a statistical framework, we assume having n measurement (x_i, y_i) $i \in \{1, \dots, n\}$ of (X, Y) . We use this data to learn the unknown relationship f between X and Y .

We can predict Y with a **predictive model** of f : \hat{f} such that:

$$\hat{Y} = \hat{f}(X).$$

According to the type of model, \hat{f} has different forms. When a model captures the information about its predictions within a finite set of parameters, it is a **parametric model**. Whereas when it doesn't require parameters it is a **non-parametric model**.

2.1.3 Goodness of predictive model

Let's introduce a function that measures the discrepancy between a response y and its prediction \hat{y} : **the loss function L** .

For the regression framework the square error loss: $L(y, \hat{y}) = (y - \hat{y})^2$ is typically used. So the **expected squared prediction error** of the predictive model \hat{f} is:

$$Err_{\hat{f}} = E \left[(Y - \hat{f}(X))^2 \right],$$

with the expectation taken over (X, Y) .

To address the goodness of \hat{f} we need to estimate the expected squared prediction error. The **empirical prediction error** is:

$$err_{\hat{f}} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2,$$

with n the number of observations. It is also called **Mean Squared Error (MSE)**.

Training (or fitting) a parametric model means finding the parameters that minimize the empirical prediction error.

An other measurement of the goodness is the **R-squared (R^2)**. It quantifies the proportion of the variation explained by the predictors among all the variation:

$$R^2_{x,y,\hat{f}} = 1 - \frac{MSE}{\text{Total sum of squares}} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{f}(x_i))^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

where \bar{y} is the mean value of the response.

2.1.4 Hyperparameter and cross-validation

We saw that the parameters are learned during the fitting. Another kind of parameter is used to control the learning process: the **hyperparameter**. As for the parameters, it needs to be tuned. **Cross-validation** is often used to this aim. It is a resampling technique that trains and tests a model at different iterations using different parts of the data.

2.1.5 Bias and variance

The expected squared prediction error for an input point $X = x_0$ can be written as:

$$Err_{\hat{f}}(x_0) = \sigma_\epsilon^2 + Bias^2 [\hat{f}(x_0)] + Var [\hat{f}(x_0)],$$

where:

- σ_ϵ^2 is the irreducible error due to the noise variable ϵ .

- The **bias** is the difference between the average prediction $E[\hat{f}(x_0)]$ and the truth at x_0 .
- The **variance** is the variability of the prediction $\hat{f}(x_0)$ when \hat{f} is fitted for different data sets.

Simple parametric methods such as linear models have low flexibility (i.e. high bias), but need fewer data to fit. There are stable across data sets (i.e. low variance).

On the other hand, complex methods such as Support Vector Regression (**SVR**), K-nearest neighbors (**KNN**), Classification And Regression Trees (**CART**), Random Forest (**RF**) and Extreme Gradient Boosting (**XGB**) *, can be very flexible (i.e. low bias), but need a lot of data. They are prone to overfit the train set and vary strongly depending on the data set (i.e. high variance).

2.1.6 Model evaluation

This subsection describes some plots that can be done to analyze the behavior of a fitted model.

Prediction error

The true outputs from the data set are plotted against the expected values produced by the model. This enables gauging the model's level of variance and, by comparing to the identity line, diagnosing the model.

Residual distribution

Residuals are the discrepancy between the true responses and the predicted ones. The residual plot displays the residuals on the vertical axis against the predictors on the horizontal axis. It allows to identify areas that may be more or less prone to mistakes and to analyze the variance. If a model can predict the output well, residuals should not show any structure,

*cf. [2.2.2](#) for the definitions

be centered at 0, and have a reasonable range. The comparison with the range of predictors addresses the reasonability of the range.

Learning curves

Learning curves plot the relationship between the training score and the test score for a given model with a different number of training samples. This plot is used to demonstrate two concepts:

1. To what extent the estimator is improved by additional data (i.e. do we have "enough data" to train the model).
2. Whether the estimator is more sensitive to bias-related error than variance-related error.

The model won't benefit from extra data if the training and cross-validation scores converge as more data is introduced. In order for the model to generalize more successfully, more training examples are likely needed if the training score is significantly higher than the validation score. There is more variability along the training score curve if the model has bias-related inaccuracy. There is more variation along the cross-validated curve if the model suffers from error due to variance.

2.2 Model implementation

2.2.1 ML workflow

Now we have all the elements needed to understand the ML framework, we explain the workflow to implement a model and compare it with other models.

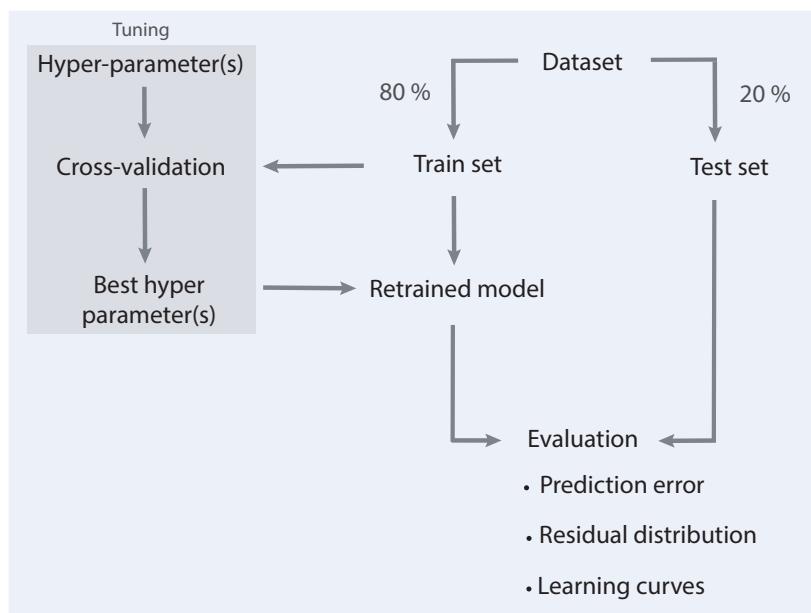
1. The data is split into training (80% of the data) and test set (20% of the data).
2. If the model has hyperparameters, they are tuned with cross-validation (cf. [2.1.4](#)).

3. The model is retrained (i.e. the parameter are fitted) with the best value of hyperparameter(s) on the train set.
4. Evaluation is performed on the test set to address the goodness of the model (cf. 2.1.6).
5. Once all models are set, their cross-validate MSE or R^2 are compared and a model is selected.

Note: Some models may need a normalization of the predictor to work well.

Figure 2.1 gives a visual representation of the steps described above.

1. Set the model



2. Comparison with other models

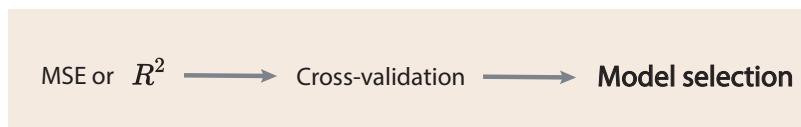


Figure 2.1: Machine learning workflow: steps to implement a given model and compare it with the others.

2.2.2 Models used in this study

Table 2.1 gives a brief definition and some characteristics of the models used for this study. More information can be found in James et al. (2013) and in Friedman et al. (2000), and Friedman (2001) for XGB.

Table 2.1: Model definition and characteristics

Model	Definition	Tuning hyperparam.	Remark
Linear	$\hat{f}(x) = \beta_0 + \sum_{i=1}^p \beta_i x_i$, with β_i parameters		Good interpretability but can be too rigid
SVR	$\min_w \frac{1}{2} \ w \ ^2$ such that $ y_i - \langle w, x_i \rangle - b \leq \epsilon$ with parameters β and ϵ	Total budget b	
KNN	$\frac{1}{k} \sum_{i; x_i \in N_k(x_0)} y_i$, with $N_k(x_0)$ the neighborhood of x_0	Number of neighbors k	Sensitive to curse of dimensionality
CART	Growing a tree by splitting the feature space recursively into high-dimensional no-overlapped rectangles that maximize the variance	Maximum depth	Easy to fit and good interpretability but prone to overfit
RF	Same process as CART but growing several decorrelated trees thanks to a constraint in the growing process	Maximum depth Number of estimators	Better performance than CART
XGB	Same process as RF but with the optimization of an arbitrary differentiable loss function	Maximum depth Number of estimators Learning rate Subsample	Better performance than RF

Chapter 3

Model implementation

All code is done in python and available [online](#).

3.1 The dataset

3.1.1 Generation of the dataset

The idea is to create a database of predictors and responses based on numerical simulations to be used in the training phase of the machine learning model. A scheme of the workflow is reported in [3.1](#).

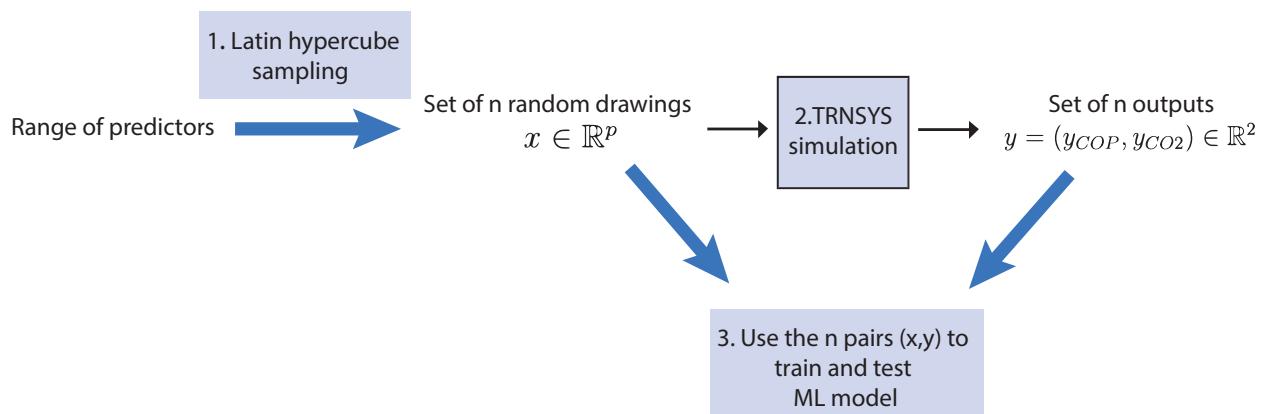


Figure 3.1: Workflow to create the dataset

To create this dataset, we proceed as follows:

1. An acceptable range of values is established for each input. Table 3.1 reports the intervals used together with the default value of each variable. We perform an Latin Hypercube Sampling (LHS)* McKay et al. (2000) on this range in order to have a set of 800 quasi-random drawings.

Table 3.1: Range of input predictors

Parameter	Minimum	Default	Maximum	Unit
DHW demand	2850	3990	5700	<i>L/day</i>
DHW secondary flow	8000	16000	20800	<i>m³/hour</i>
DHW recirculation flow	400	800	1600	<i>m³/hour</i>
Temperature cold water	5	10	15	°C
DHW volume tank	1	1.9	9	<i>m³</i>
SH volume tank	1	2.2	9	<i>m³</i>
SH secondary flow	8000	13000	16000	<i>m³/hour</i>
SH fraction compensation	0.1	0.3	0.4	%
Temperature boiler room	15	18	23	°C
Tank loss coefficient	0.038	0.4	0.8	<i>W/m²K</i>
DHW pipe loss coefficent	4	8	12	<i>W/m²K</i>
SH-HP heating curve	0	1	5	°C
SH tank heating curve	-5	0	5	°C
DHW heat exchanger effectiveness	0.7	0.8	0.9	no unit
DHW set point	40	45	50	°C

2. We use these drawings as input of the numerical simulation. Each simulation takes about 4 minutes, so the 800 simulations take about 53 hours. The outputs are the COP and the *CO₂* emitted. The simulation also provides information about the convergence of the process. Simulation that doesn't converge is discarded. There are 13 runs of this kind over the 800 simulations.

3. The 800 pairs of input-output are used to train and test ML models.

3.1.2 Inputs distribution

Figure 3.2 shows the normalized input distribution for each predictor. Normalization is performed since the range of each predictor is very different. The distributions are quite uniform across the given range, so LHS worked well. Note that the default value is not especially at the center of the range.

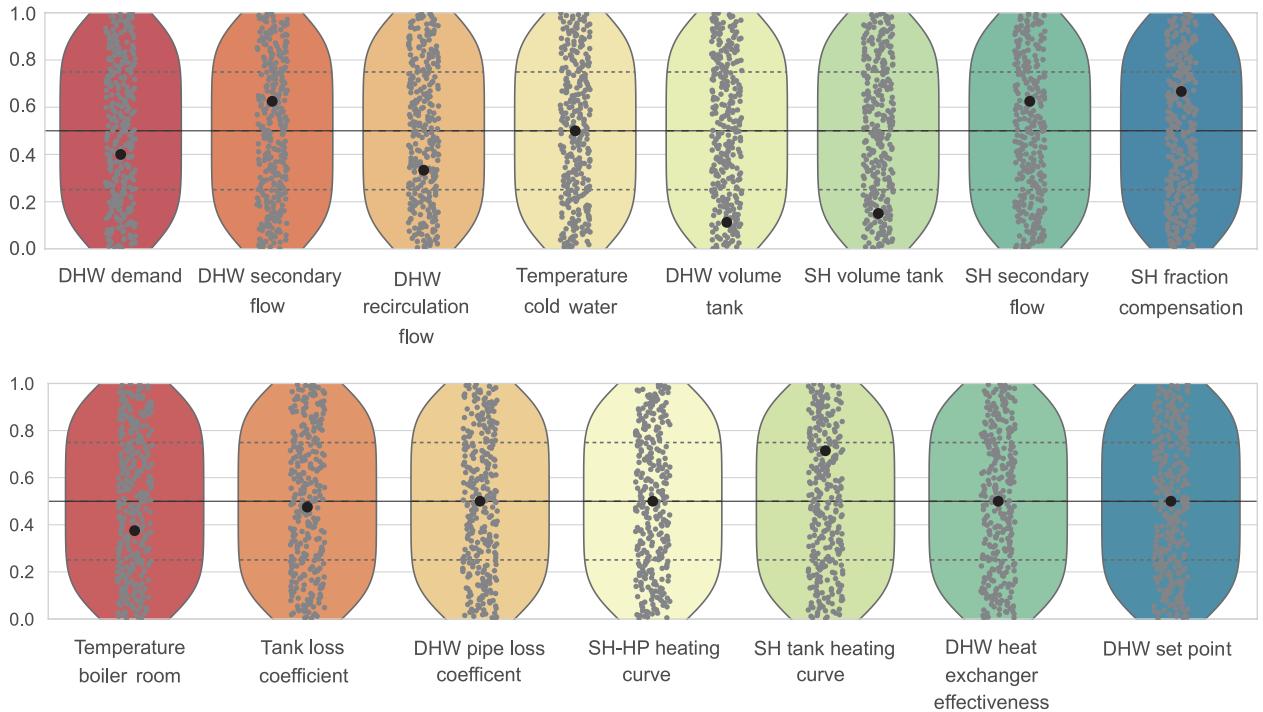


Figure 3.2: Input normalized distributions for each predictor. Each grey dot is an input. Black dots are the normalized default values.

3.1.3 Outputs distribution

We will analyze the outputs of the dataset. First we look into each output distribution and then we examine the relationship between them.

Figure 3.3a shows the distribution of the COP. The range is from 2.73 to 3.08 with a mean of 2.91. Figure 3.3b shows the distribution of the CO_2 emitted. The range is from 20.94 to 26.20 with a mean of 23.21.

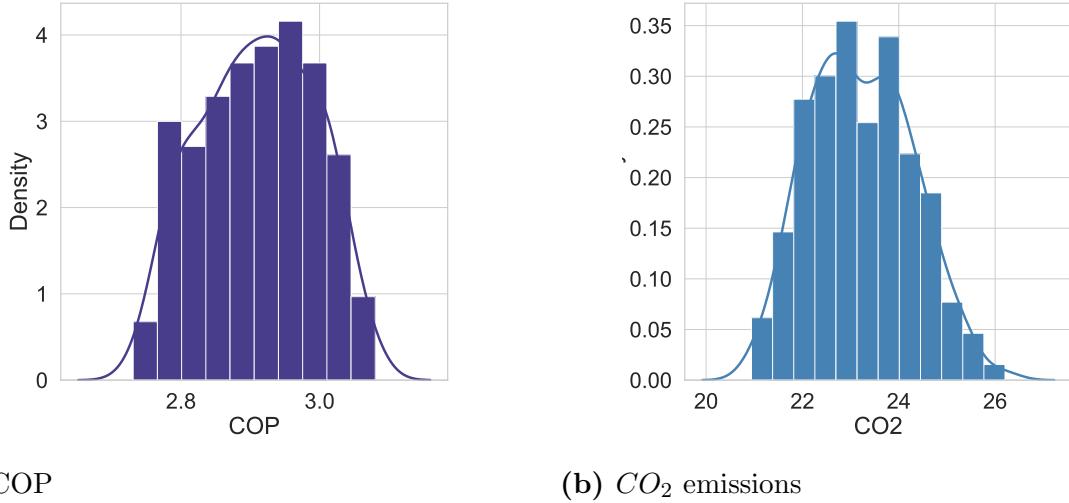


Figure 3.3: Outputs distribution: COP and CO_2 emissions.

These two distributions seem to follow a normal distribution.

3.1.4 Correlation between outputs

Let's compute the correlation matrix of the outputs (Figure 3.4).

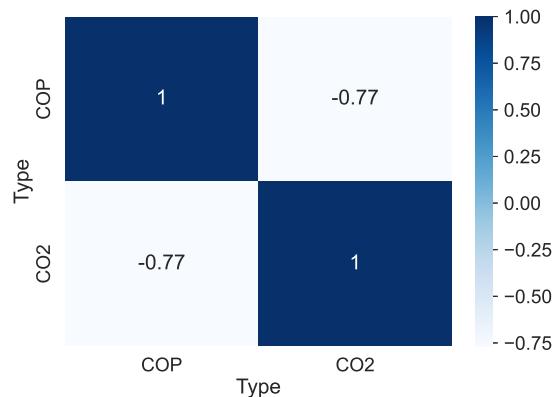


Figure 3.4: Correlation matrix of the simulation outputs

There is a negative correlation between the COP and the CO_2 emissions: -0.76.

Figure 3.5 shows the distribution of the COP against the CO_2 emissions. The negative correlation between these two outputs is clearly visible.

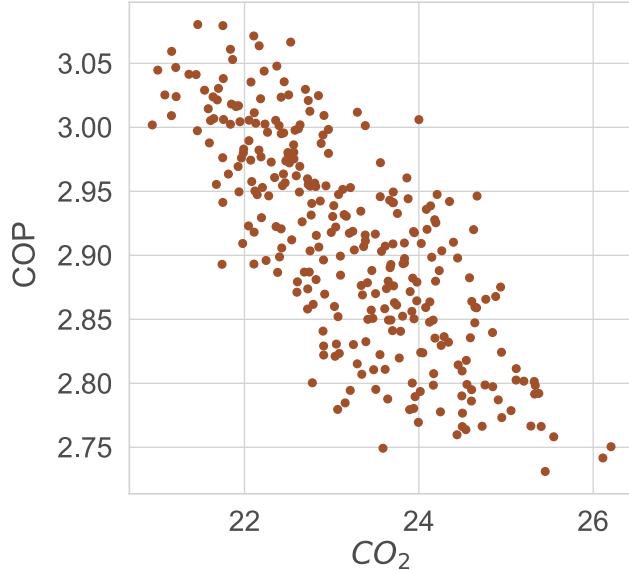


Figure 3.5: Distribution of the COP against the CO_2 emissions

3.2 Model implementation

In this section, we train some ML models and analyze them with the prediction error plot, the residual distribution plot and the learning curves plot. Explanations of these methods and their interpretation are provided in Chapter 2. All model implementation is done with Scikit-Learn [Pedregosa et al. \(2011\)](#), and all plots use Yellowbrick [Yellowbrick \(2022\)](#).

Let's begin with the simplest model: linear model.

3.2.1 Linear model

Prediction error

Figure 3.6 shows the prediction error of the linear model for the COP and CO_2 emissions. The plot shows a small variance, and a very large bias. Indeed, the best fit is the vertical

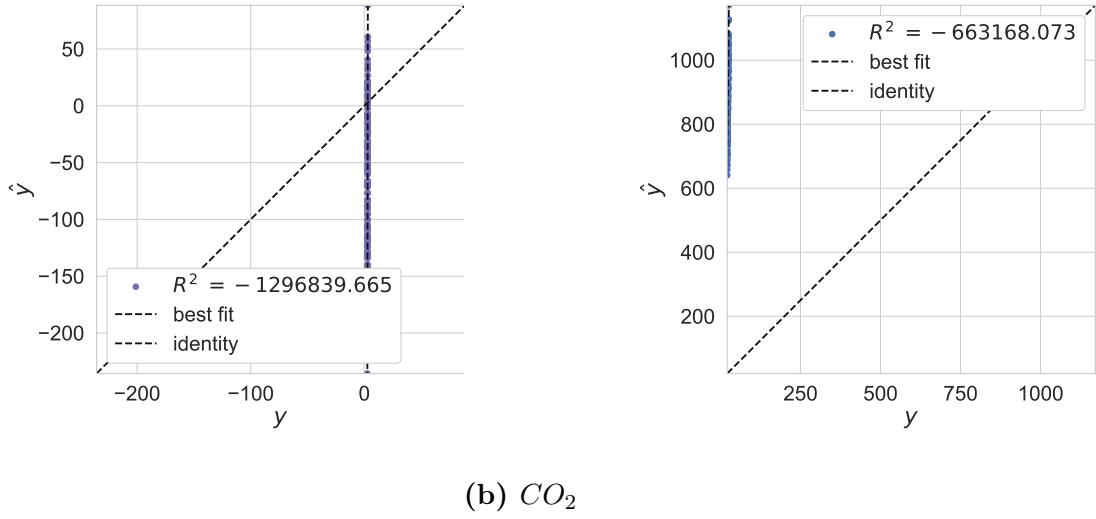


Figure 3.6: Prediction error of the linear model

line which is very different of the identity line. This hints us that this model is too rigid to be able to fit complex data (in agreement with the theory 2.1.5 and 2.1). Let's look at the other plots to confirm our observation.

Residual distribution

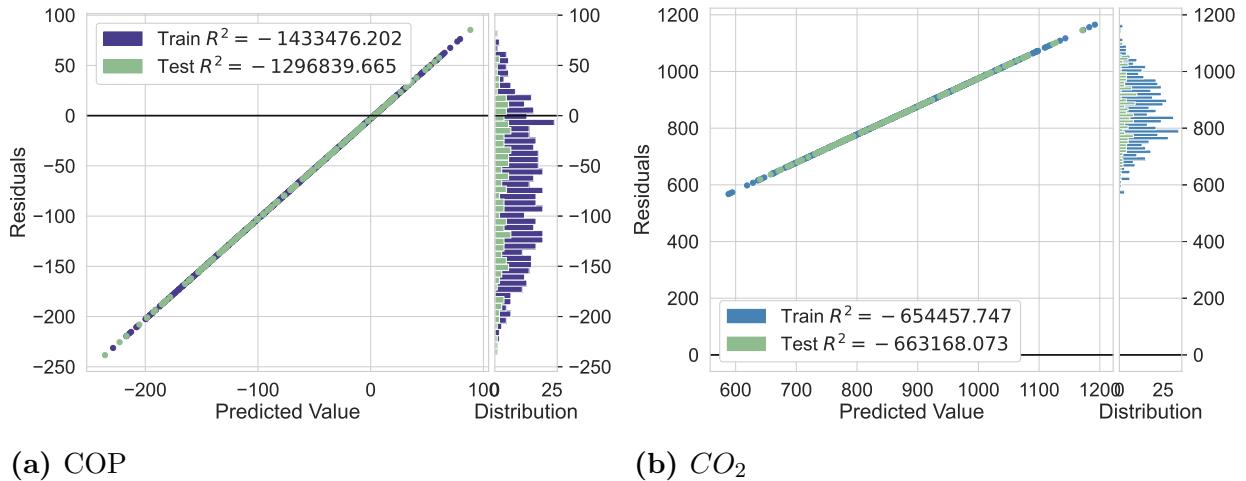


Figure 3.7: Residual distribution of the linear model

Figure 3.7 shows the residual distribution of the linear model for the COP and CO_2 emissions. The trained and tested R^2 is very negative for both COP and CO_2 . Moreover, the residuals

increase with the predicted values. So the residuals still have a structure and the linear model is not able to explain all the complexity of the data.

Cook distance

When conducting a linear regression analysis, Cook's distance can be used to estimate the influence of a data point on the fitting and thus detect outliers. The threshold commonly used is $4/n$ with n the number of observations in the dataset.

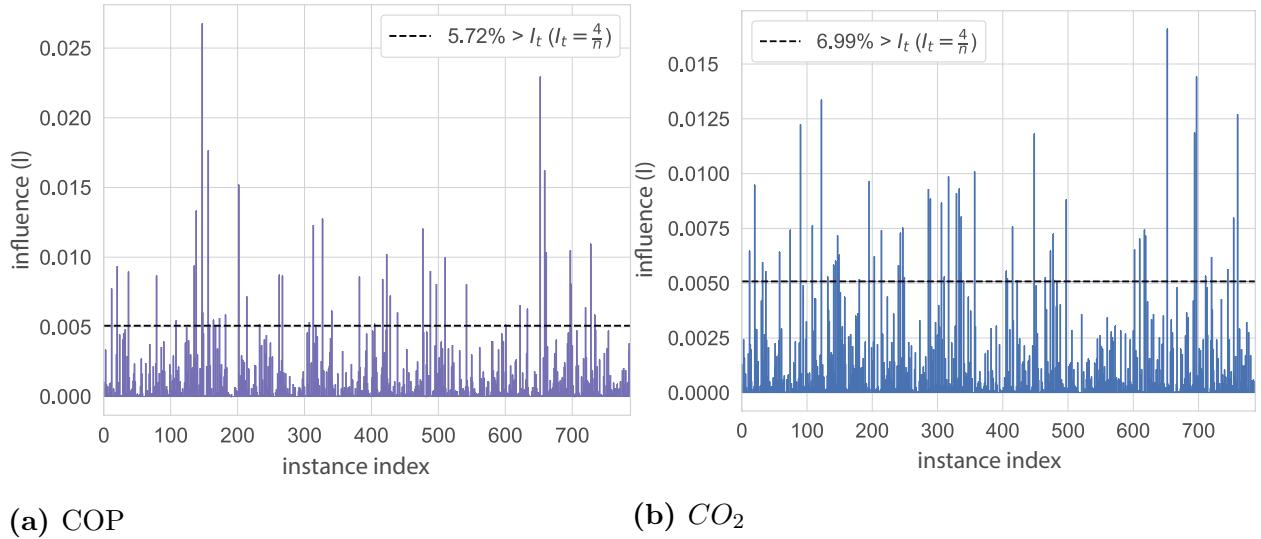


Figure 3.8: Outlier detection of the linear model with Cook's distance

Figure 3.8 shows the Cook distance plot for the linear model. About 6% of the data for the COP and 7 % of the data for the CO_2 are considered outliers. These values are quite large.

All previous analysis show that this model is too rigid for our complex data.

3.2.2 More complex models

Since the linear model is too simple to predict the outputs well, we implement the following high-complexity models: **SVR**, **KNN**, **CART**, **RF** and **XGB** (cf 2.1 for a description of these models). This section provides an analysis of each model (i.e. hyperparameter selection, prediction error, residual distribution, and learning curves).

Table 3.2, Table 3.3, Table 3.4 and Table 3.5 resum the results of these analysis. A comparison between model can be found in the next section.

Hyperparameter selection

In Appendix, Figure 6.2, Figure 6.3, Figure 3.11, Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7, Figure 6.8 and Figure 6.9 show the hyperparameter selection for the COP and CO_2 emissions for SVR, KNN, CART, RF and XGB, respectively.

Prediction error

Figure 3.9, Figure 3.10, Figure 3.11, Figure 3.12, and Figure 3.13 show the prediction error for the COP and CO_2 emissions for SVR, KNN, CART, RF and XGB respectively. We keep the same scale for the COP and CO_2 across models be facilitate the comparison.

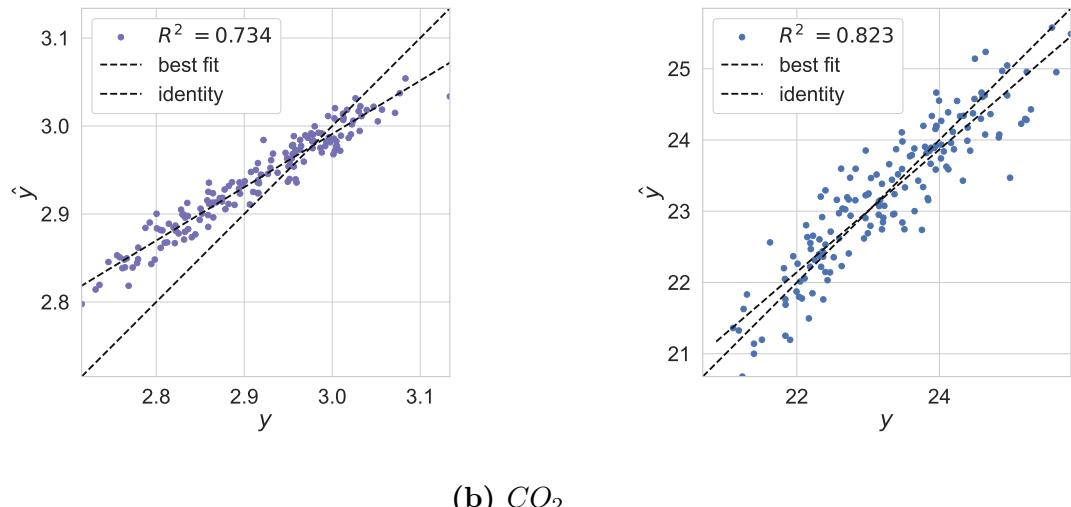


Figure 3.9: Prediction error for the SVR

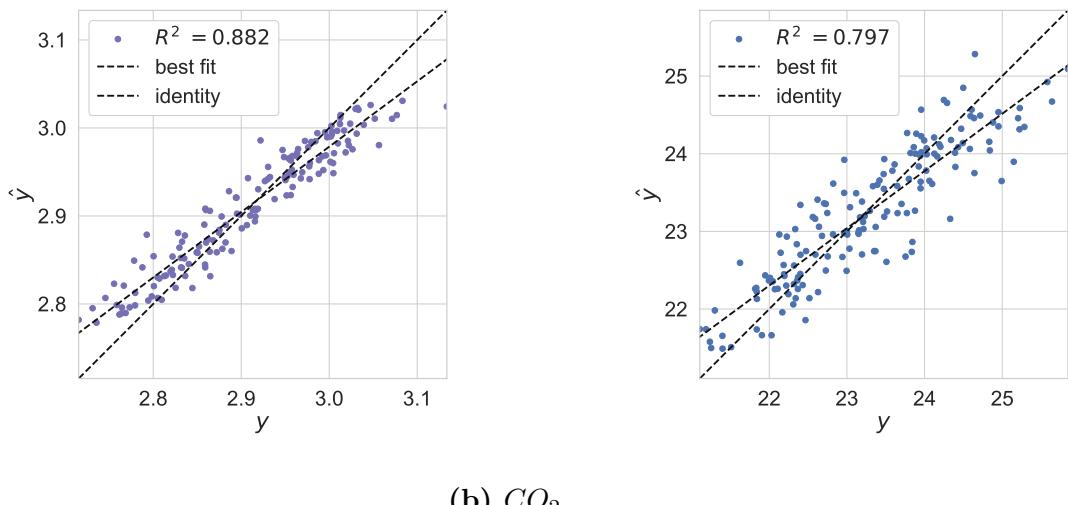


Figure 3.10: Prediction error for the KNN

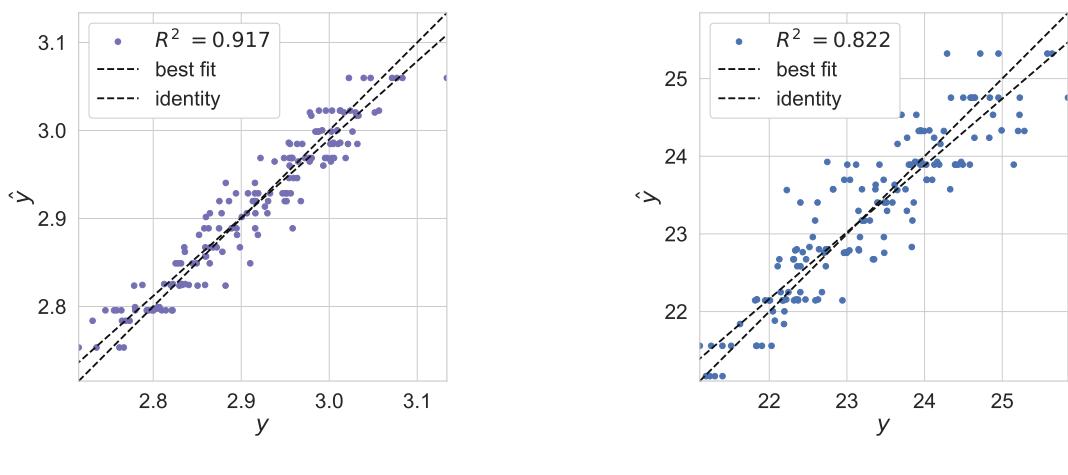


Figure 3.11: Prediction error for the CART

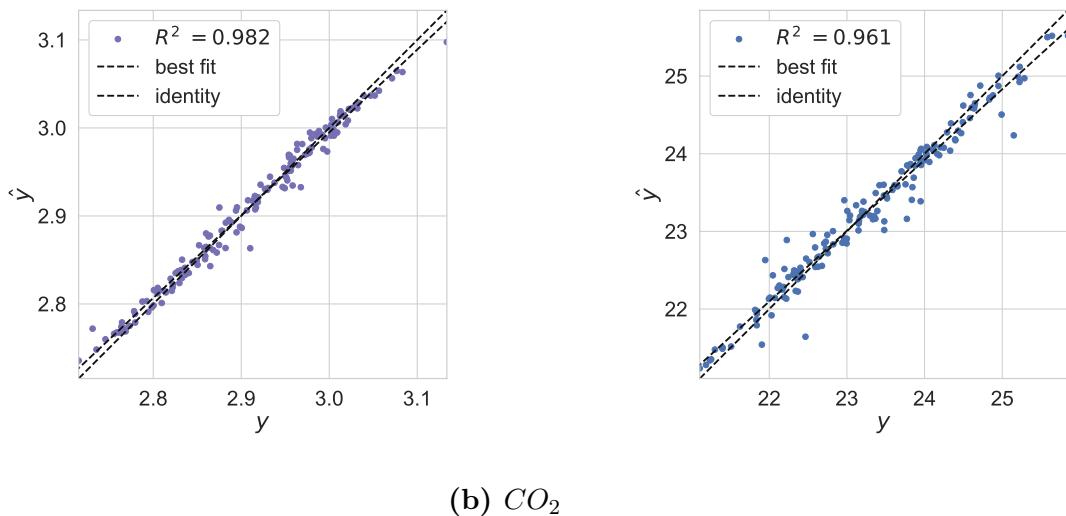


Figure 3.12: Prediction error for the RF

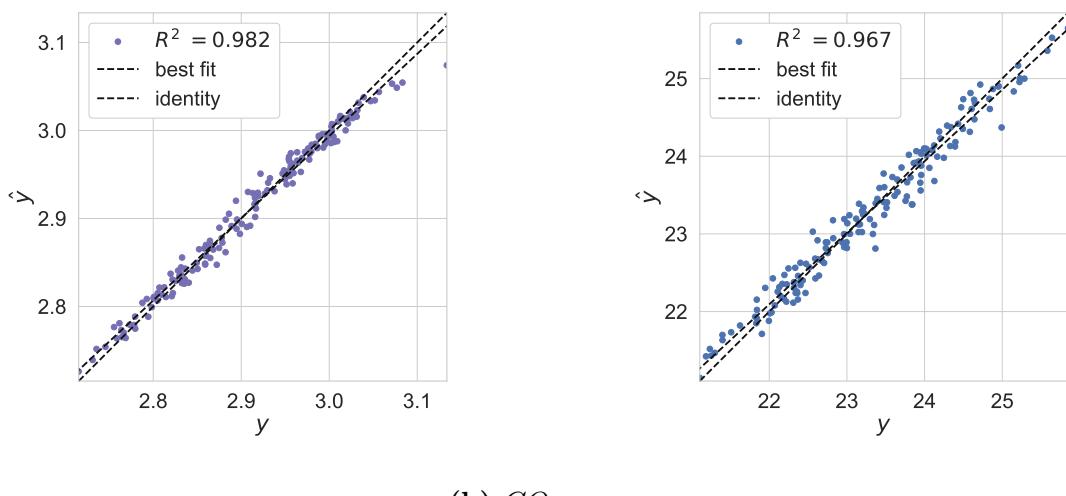


Figure 3.13: Prediction error for the XGB

Table 3.2: Results of the predicted error plots

Model	Pos. fit COP	Var. COP	Pos. best fit CO_2	Var. CO_2
SVR	very far	large	far	large
KNN	far	large	far	large
CART	close	large	far	large
RF	close	small	close	small
XGB	close	small	close	small

Table 3.2 resums the results of the predicted error plots. RF and XGB seem to have better performance than the other models.

Residual distribution

Figure 3.14, Figure 3.15, Figure 3.16, Figure 3.17, and Figure 3.18 show the residual distribution for the COP and CO_2 emissions for SVR, KNN, CART, RF and XGB respectively.

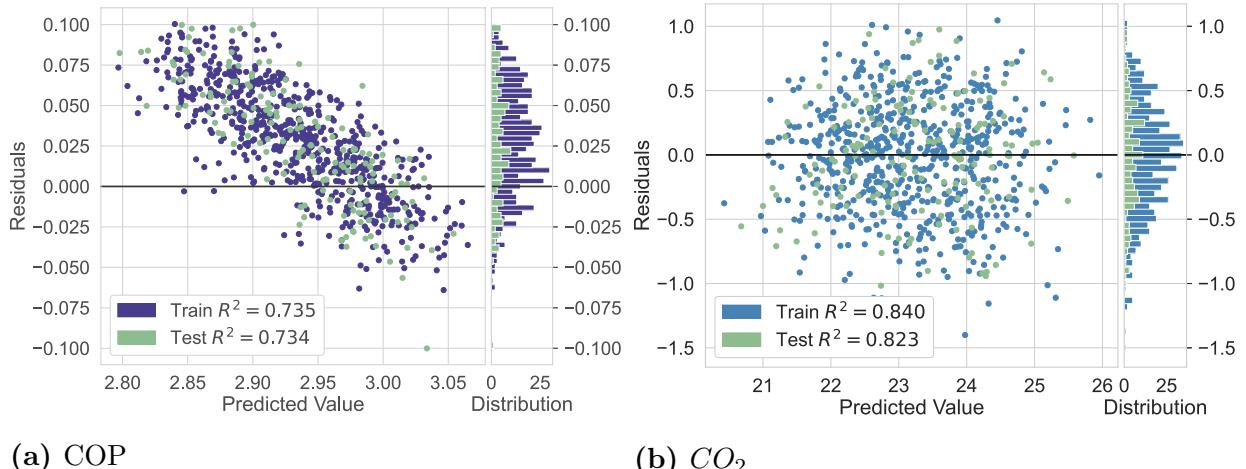


Figure 3.14: Residual distribution for the SVR

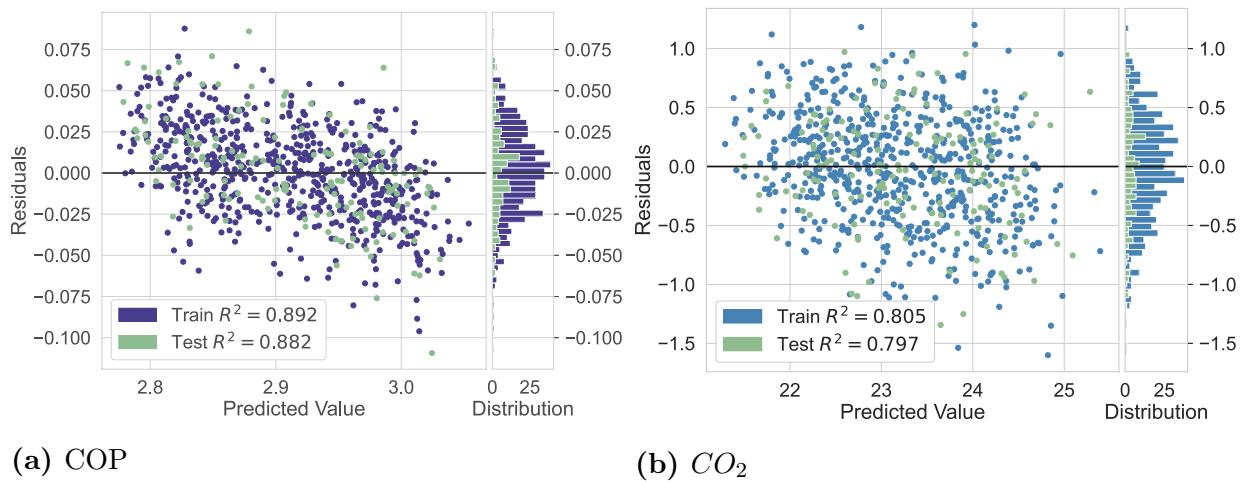


Figure 3.15: Residual distribution for the KNN

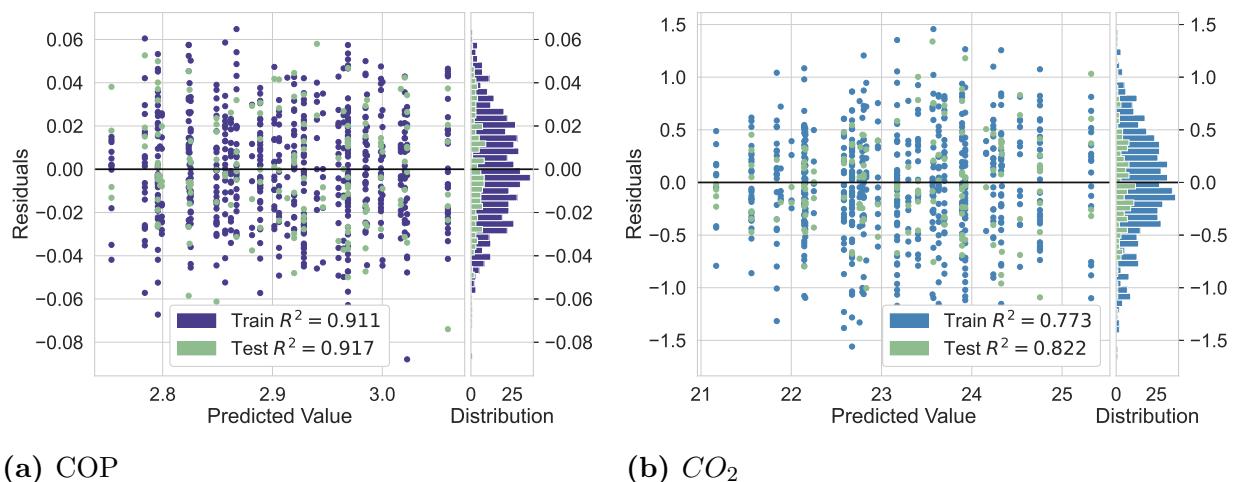


Figure 3.16: Residual distribution for the CART

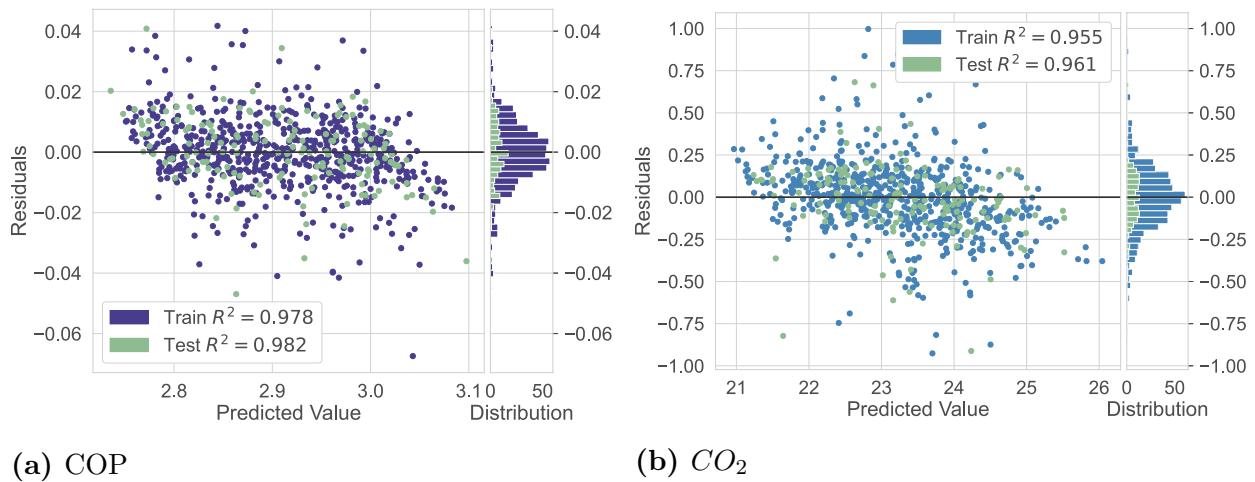


Figure 3.17: Residual distribution for the RF

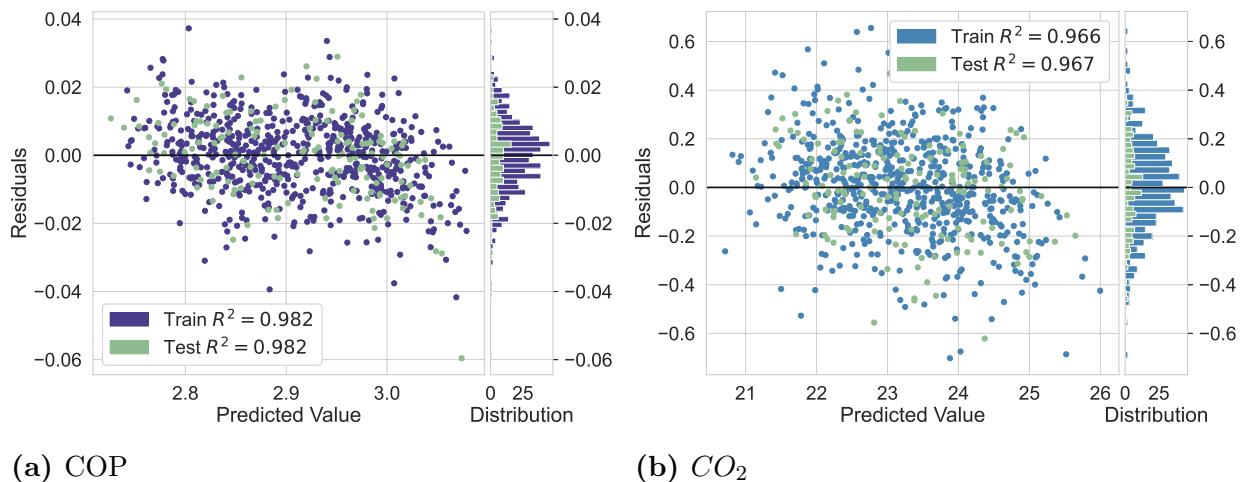


Figure 3.18: Residual distribution for the XGB

Table 3.3 resums the results of the predicted error plots. XGB seems to have better residual distribution than the other models.

Table 3.3: Results of residual distribution plots

Model	Bias COP	Var. COP	Bias CO_2	Var. CO_2
SVR	large	large	small	large
KNN	medium	small	small	large
CART	small	small	medium	large
RF	small	small	medium	medium
XGB	small	small	small	small

Learning Curves

Figure 3.19, Figure 3.20, Figure 3.21, Figure 3.22, and Figure 3.23 show the learning curves for the COP and CO_2 emissions for SVR, KNN, CART, RF and XGB respectively.

The curves are plotted with the mean scores however variability during cross-validation is shown with the shaded areas that represent a standard deviation above and below the mean for all cross-validations. Cross validation is performed with 5 folds and 3 repetitions.

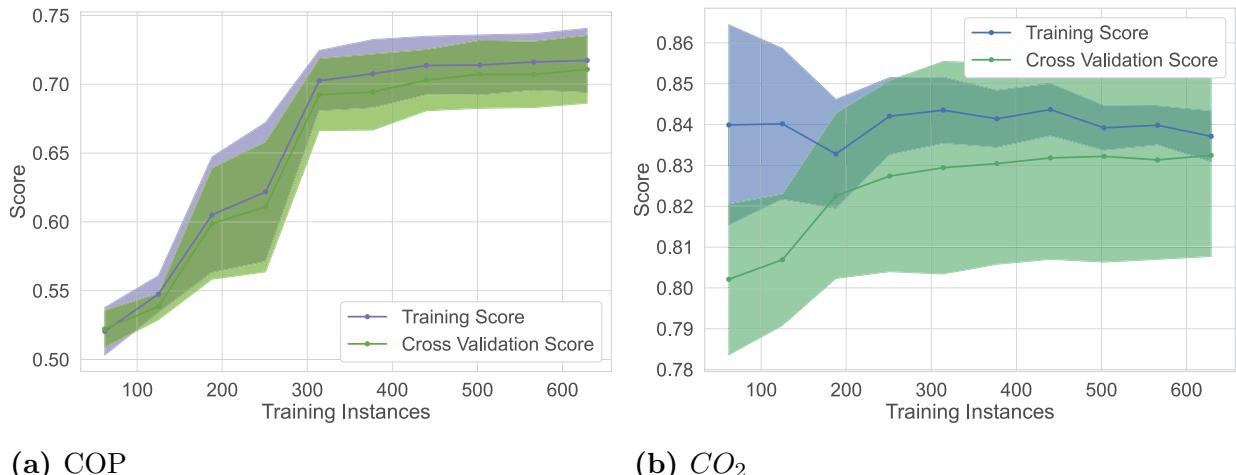


Figure 3.19: Learning curves for the SVR

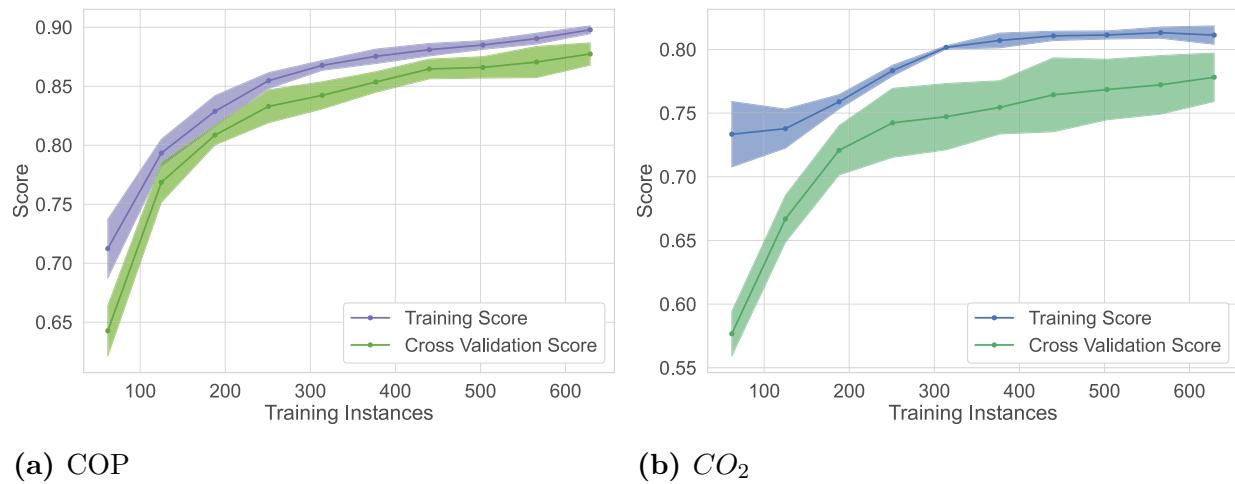


Figure 3.20: Learning curves for the KNN

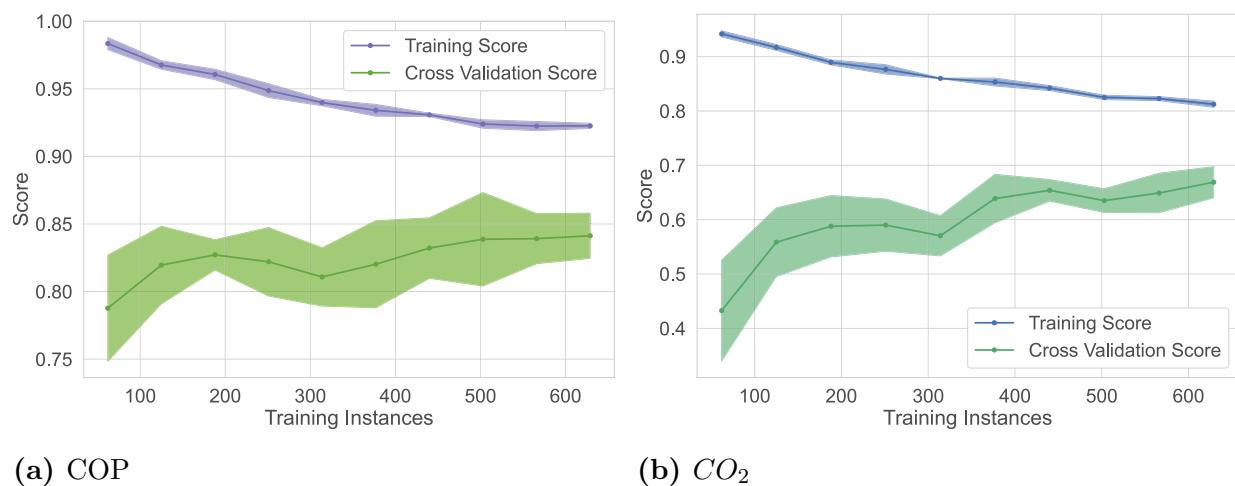


Figure 3.21: Learning curves for the CART

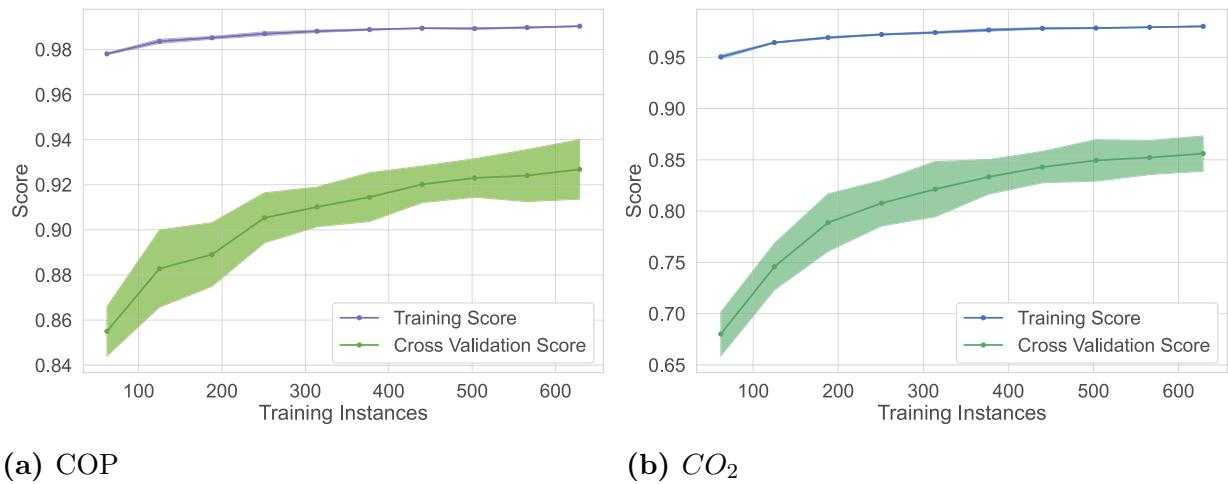


Figure 3.22: Learning curves for the RF

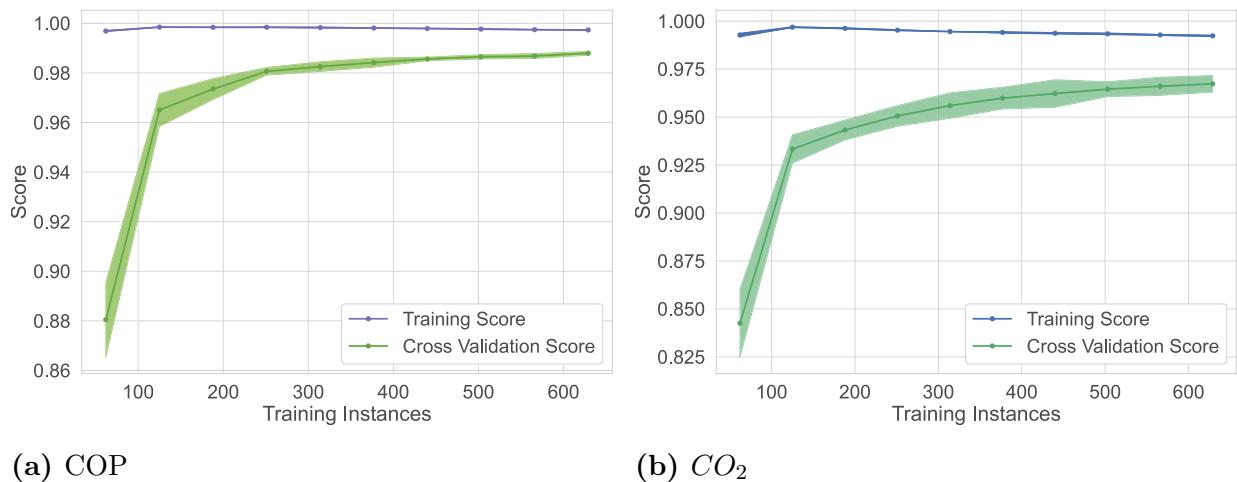


Figure 3.23: Learning curves for the XGB

The interpretation of these plots is gathered in Table 3.4 and Table 3.5.

Table 3.4: Results of COP learning curves plots.

Model	Relative position	Train var	Test var	Interpretation
SVR	together from the beginning	large	large	large bias and variance
KNN	close from the beginning	small	small	
CART	more data to converge	small	medium	relative variance need more data
RF	more data to converge	very small	medium	relative variance need more data
XGB	converge together later	none	small	need a lot of data

Table 3.5: Results CO_2 emissions learning curves plots.

Model	Relative position	Train var	Test var	Interpretation
SVR	close from the beginning	large	large	large bias and variance
KNN	close from the beginning	small	medium	large variance
CART	more data to converge	small	medium	relative variance need more data
RF	more data to converge	very small	medium	relative variance need more data
XGB	converge together later	none	small	need a lot of data

3.3 Model comparison

In order to compare the performance of the implemented models, we compute the 5-folds cross-validate R^2 score on training and test set. The same analysis but with the MSE can be found in appendix 6.3.

Figure 3.24 and Figure 3.25 plot the distribution of the COP R^2 on the cross-validated train and test set respectively. Figure 3.26 and Figure 3.27 plot the distribution of the CO_2 R^2 on the cross-validated train and test set respectively.

Note. Closer the score is to 1, better is the performance. If the test R^2 score is much lower than the training R^2 score, the model experiment overfit: it becomes too specific to the training set and it is no able to generalize well.

3.3.1 COP

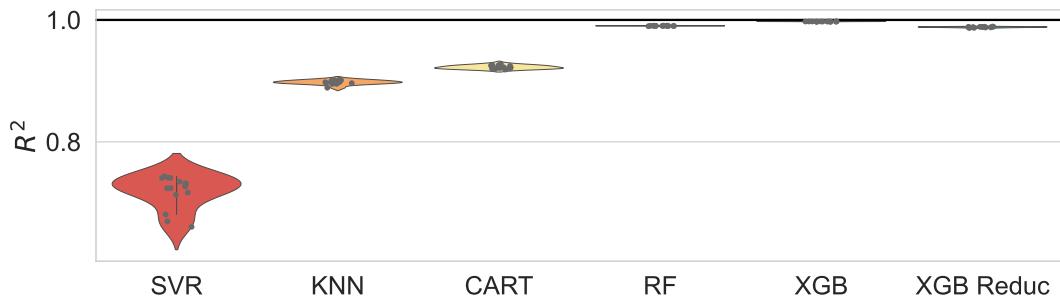


Figure 3.24: Model comparison with violin plot of the COP R^2 on train set. 5 folds CV, 3 repetitions.

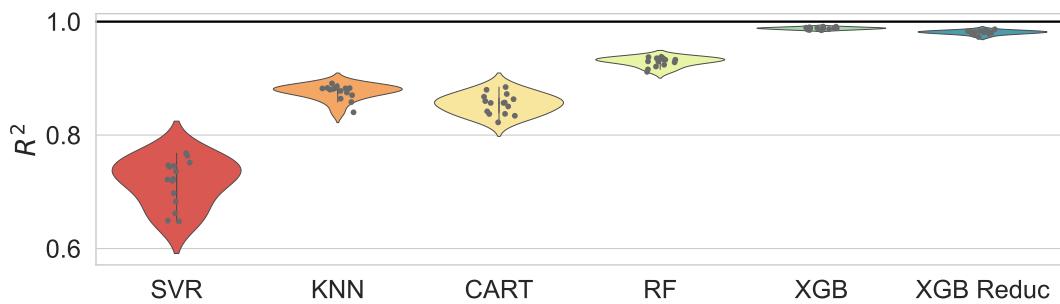


Figure 3.25: Model comparison with violin plot of the COP R^2 on test set. 5 folds CV, 3 repetitions.

- SVR performance is the lowest.

- There is not big changes between train and test R^2 score for KNN but overall the performance is low.
- CART R^2 score is much lower and broad in the test set than in the training set exhibiting overfit and high variance.
- RF has a better performance than the previous models but its R^2 score is much lower and broad in the test set exhibiting overfit and high variance.
- XGB has a very good performance for the training set as well as the test set and not increase of the variance in the test set.

3.3.2 CO_2

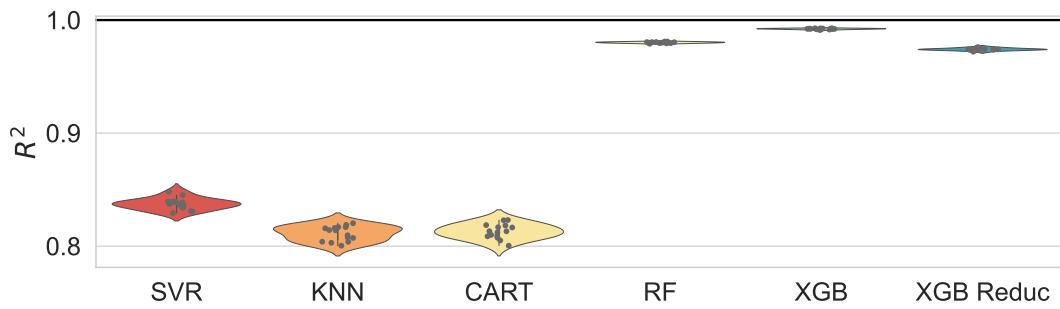


Figure 3.26: Model comparison with violin plot of the CO_2 R^2 on train set. 5 folds CV, 3 repetitions.

Note 1 Figure 3.26 and Figure 3.27 have not the same scale.

Note 2 The CO_2 emissions seem more difficult to predict since all model have a lowest performance.

- SVR, KNN and CART have a low performance in the training set and it is even lower in the test set exhibiting an large bias.
- RF has a better performance than the previous models but its R^2 score is much lower and broad in the test set exhibiting overfit and high variance.

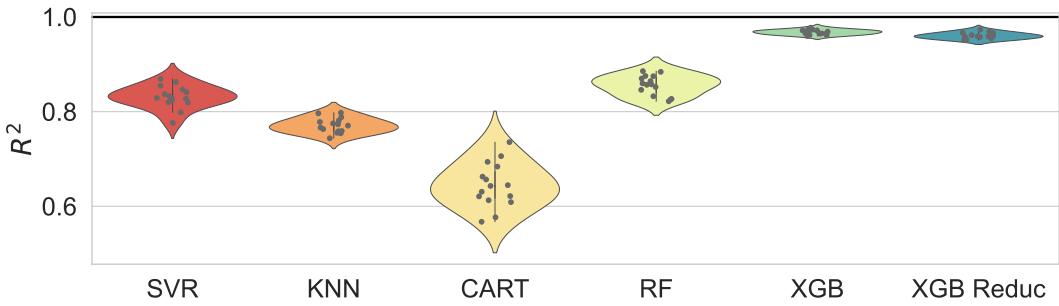


Figure 3.27: Model comparison with violin plot of the $CO_2 R^2$ on test set. 5 folds CV, 3 repetitions.

- XGB has a very good performance for the training set as well as the test set and not increase of the variance in the test set.

These results are in agreement with the previous analysis (in 3.2.2).

3.3.3 Summary and model selection

Table 3.6: Summary of the model analysis

Model	Pros	Cons	Performance
Linear model		Too rigid	Bad
More complex models (KNN, SVR, CART, RF)	Flexible Can fit complex data	Need a lot of data High variance	Medium
XGB	Can fit very complex data without overfitting or high variance	Need a lot of data	Excellent

These results are in agreement with the theory 2.1.5 and 2.1.

The model with the highest performance and a good balance bias-variance for both the COP prediction and CO_2 emissions prediction is **Extreme Gradient Boosting**. We choose this model to run the optimisation.

Chapter 4

Final model and optimization

4.1 Final model implementation

To be able to perform the prediction, we reduce the dimension of the predictors . To do so, we need to know how many predictors we should keep to still have a good performance (at least 95 %), and which ones.

4.1.1 Features selections

Recursive feature elimination

Let's address the first part of this problem with the “recursive feature elimination” method.

The **recursive feature elimination** is a feature selection approach that fits a model and eliminates the weakest features until the number of features requested is attained. It aims to minimize dependencies and collinearity in the model by iteratively deleting a small number of features every loop. Cross-validation is used to score several feature subsets and pick the top scoring collection of features to determine the optimal amount of features.

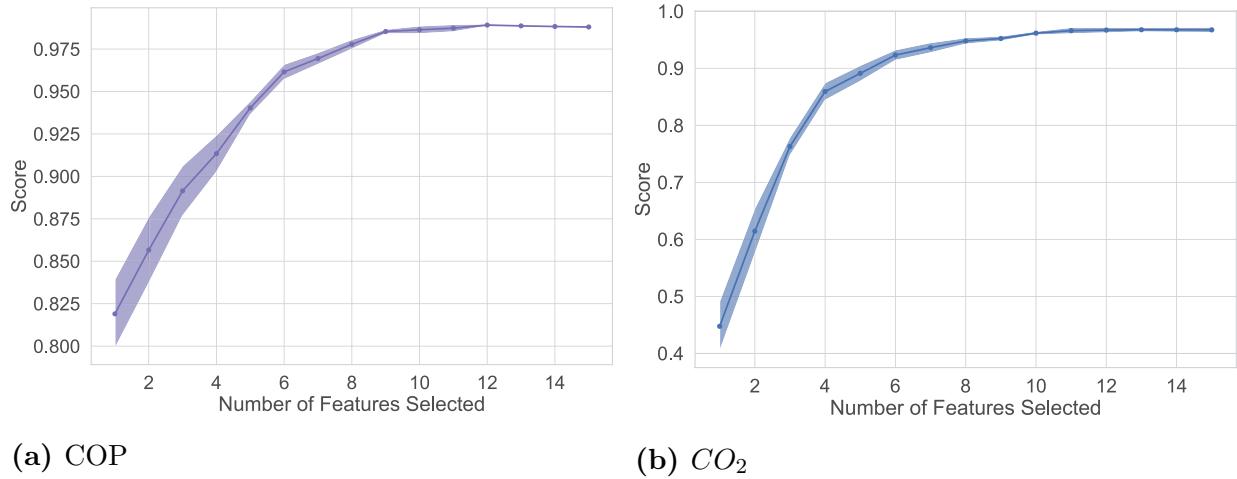


Figure 4.1: Recursive feature elimination for XGB

Figure 4.1 shows the cross-validated test score and variability according to the number of features in the model. The curves leap to high accuracy and then stabilize when the non-informative features are added to the model. It comes out that six features allowed us to reach about $R^2 = 0.95$ for the COP and CO_2 emissions.

In the next section, we address the importance of each feature for XGB, to know which of these six predictors we need to keep.

Feature importance

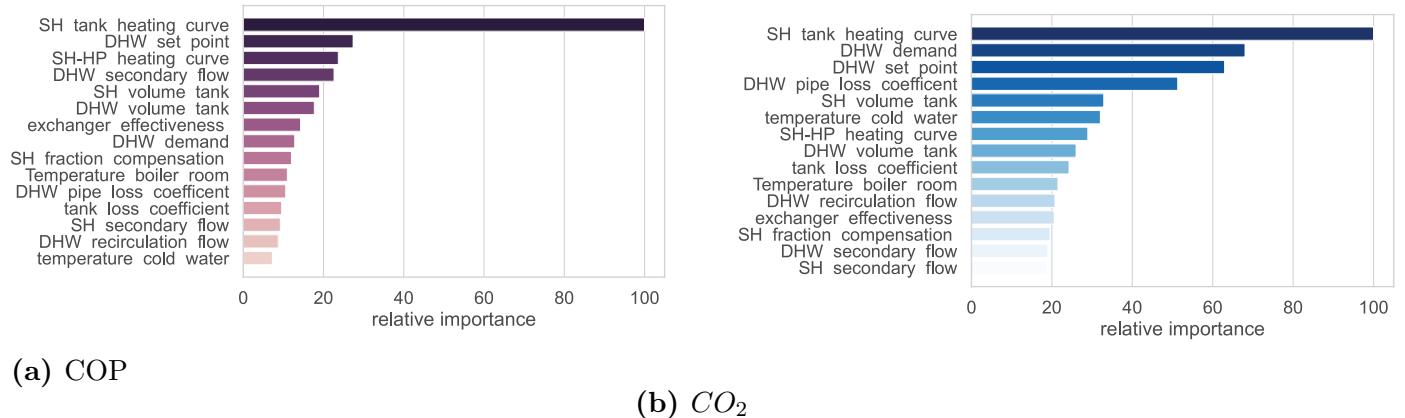


Figure 4.2: Features ranked by the explained variance that each feature adds to the model

Figure 4.2 displays the features ranked by the explained variance that each feature adds to the model. They are plotted according to their relative importance (i.e. the percent importance of the most important features).

We choose to select the six predictors listed below based on the findings of the previous features analysis: **DHW demand**, **DHW secondary flow**, **SH volume tank**, **SH HP heating curve**, **SH tank heating curve**, and **DHW set point**.

4.1.2 Reduced XGB implementation and analysis

Implementation

Figure 4.3 shows the final model implementation workflow. After the extraction of the most relevant predictors based on the XGB model, we tune, train and test a XGB model with these predictors only.

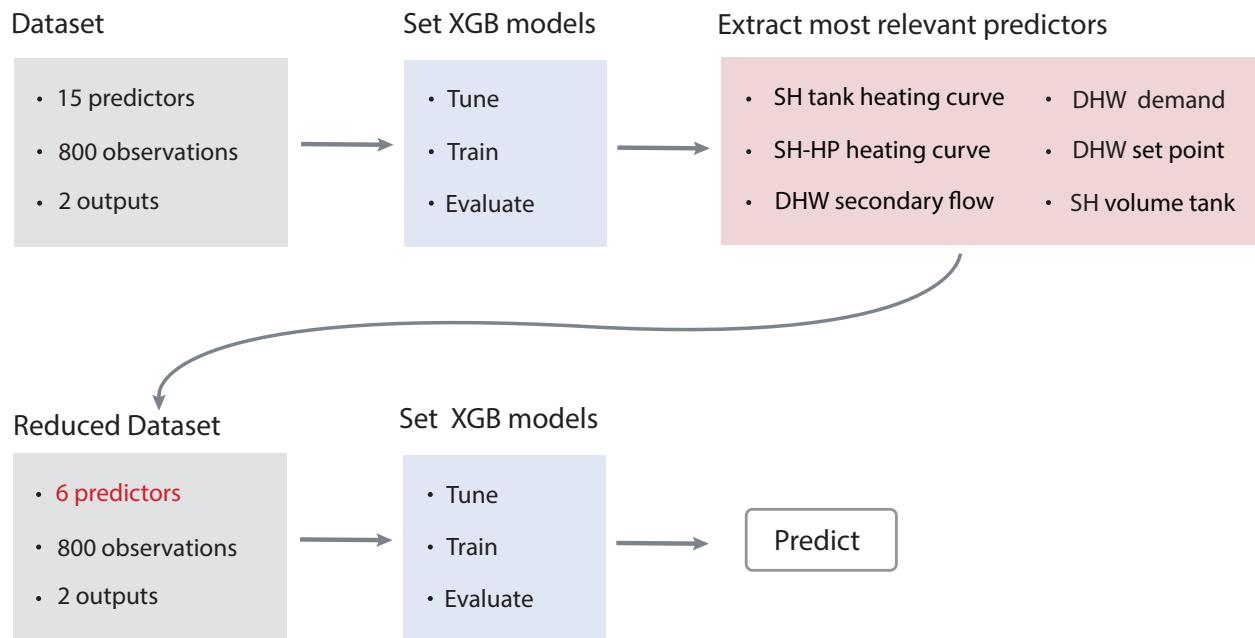


Figure 4.3: Final model implementation workflow

Hyperparameter selection

Figure 4.4, Figure 4.6, Figure 4.6 and Figure 4.7 show the final model hyperparameter selections for the COP and CO_2 emissions.

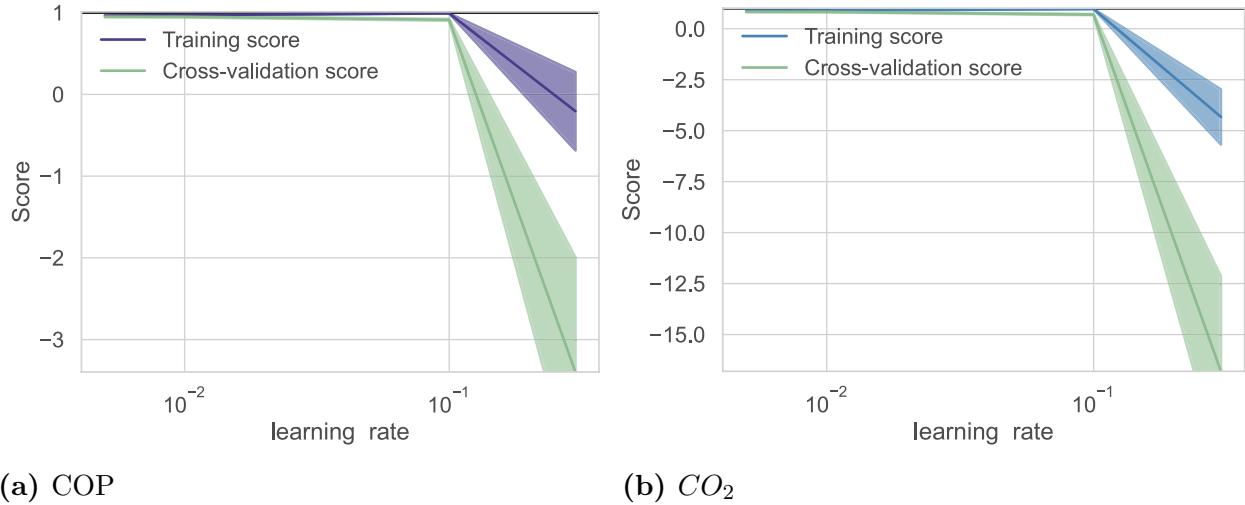


Figure 4.4: Learning rate selection for the final model

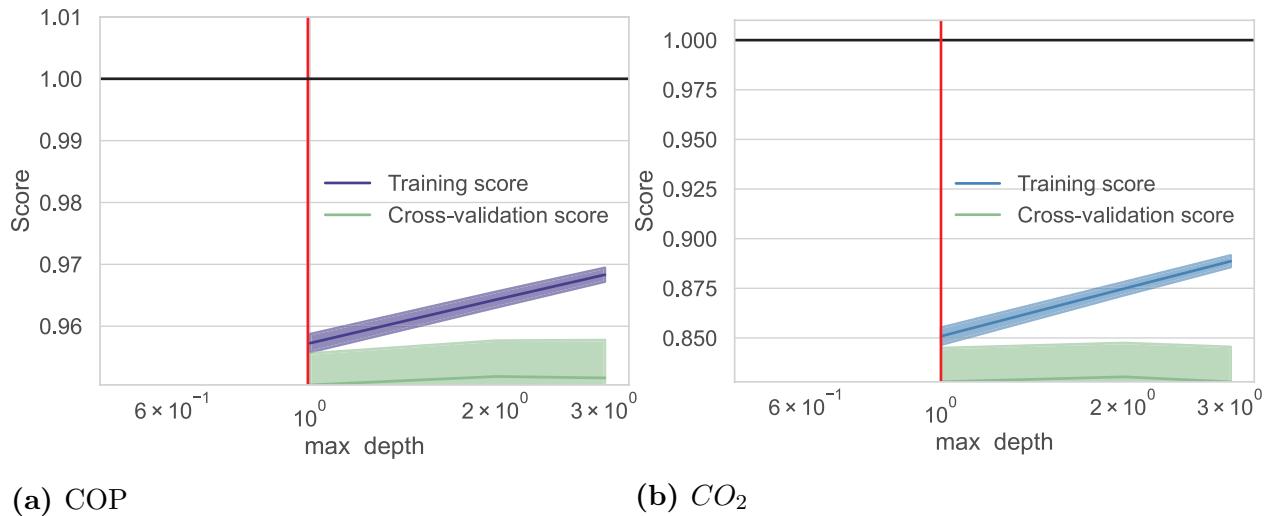


Figure 4.5: Maximum depth selection for the final model

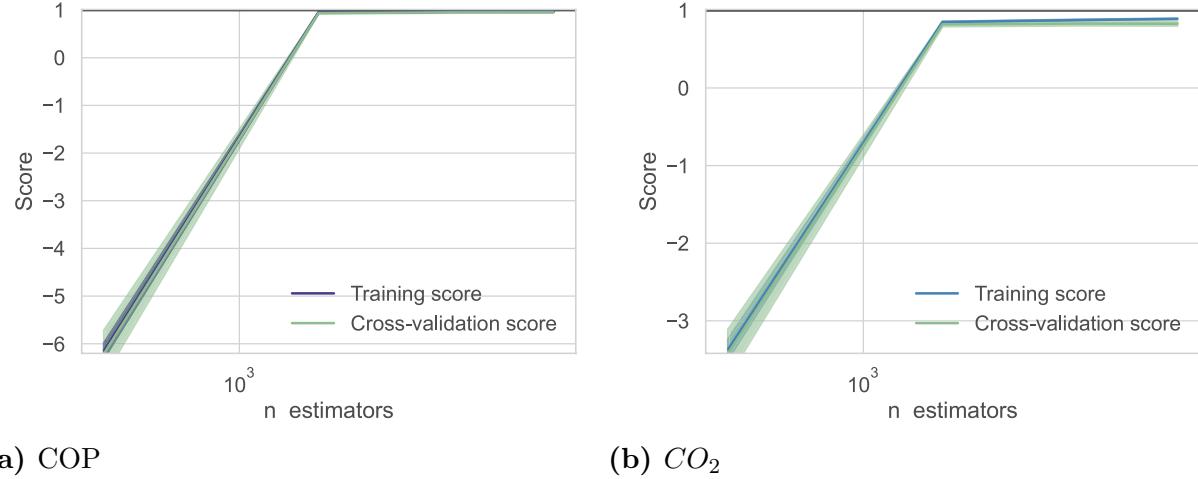


Figure 4.6: Number of estimators selection for the final model

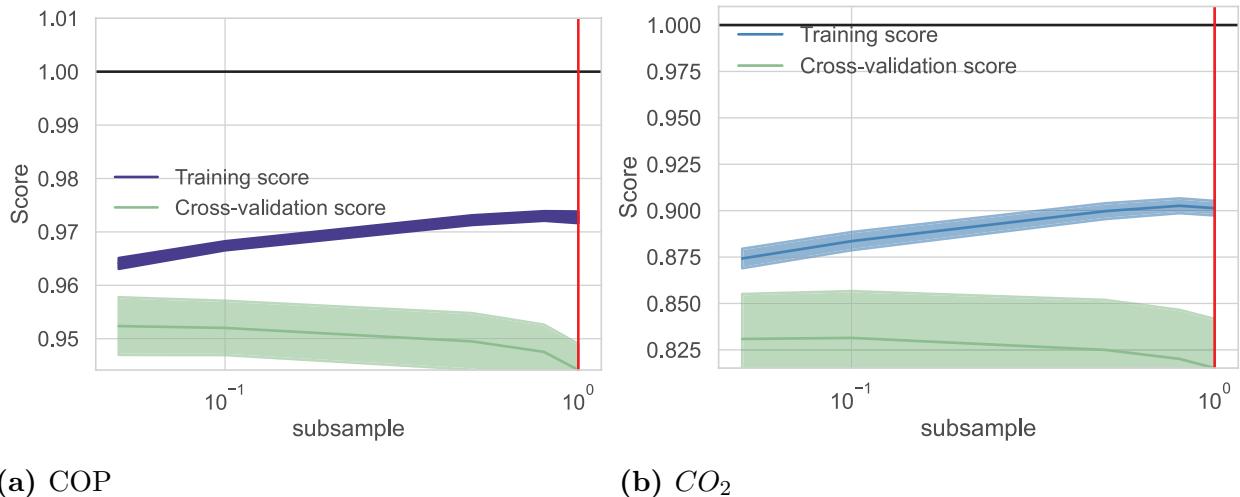


Figure 4.7: Subsample selection for the final model

Prediction error

Figure 4.8 shows the prediction error. The R^2 of the COP is very good (0.930), and the best fit line is close to the identity line. On the other hand the R^2 of the CO_2 emission is less good (0.803), but still correct, and the best fit line is further from the identity line.

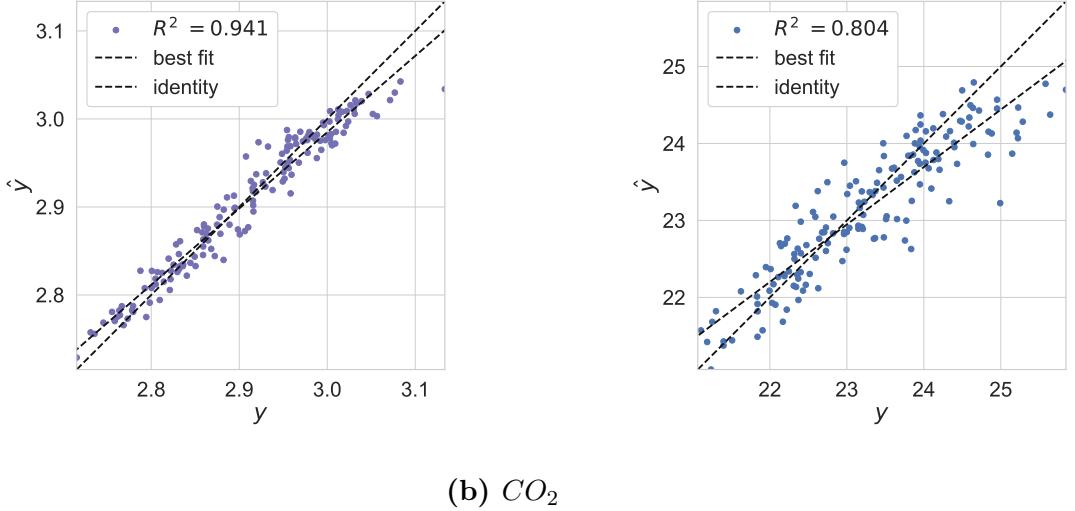


Figure 4.8: Prediction error for the final model

Residual distribution

Figure 4.9 shows the residual distribution for the COP and CO_2 emissions. Both don't have a specific structure, implying a low bias. However, the variance of the CO_2 emissions is quite large.

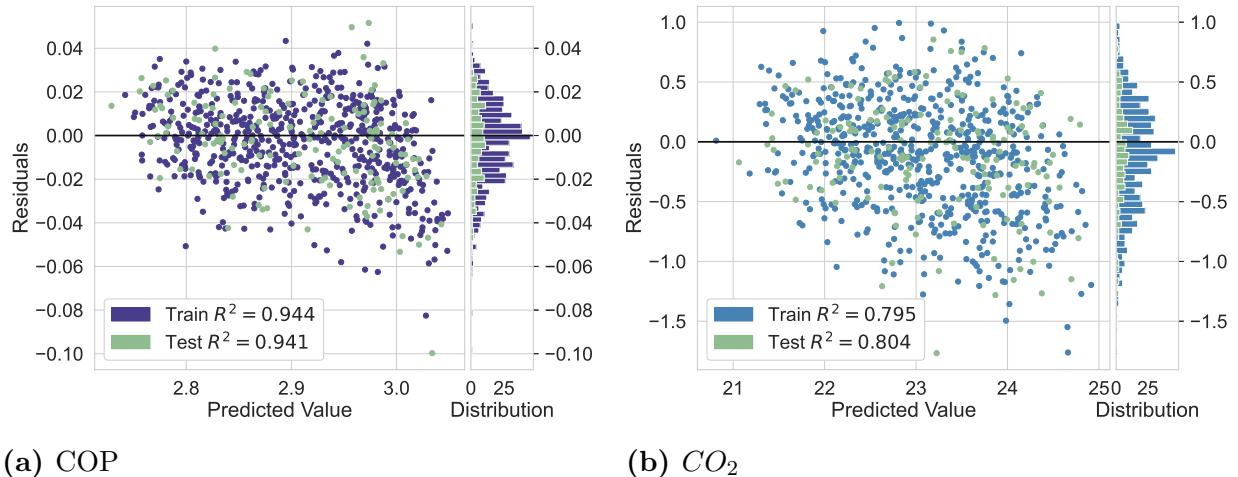


Figure 4.9: Residual distribution for the final model

Learning curves

Figure 4.10 shows the learning curves for the COP and the CO_2 emissions.

For the COP plot, the training and test curves are close from the beginning. The variabilities are small, so the bias and variance should be small. For the CO_2 plot, the training and test curves get closer for large training instances. The variability of the test curve is quite large.

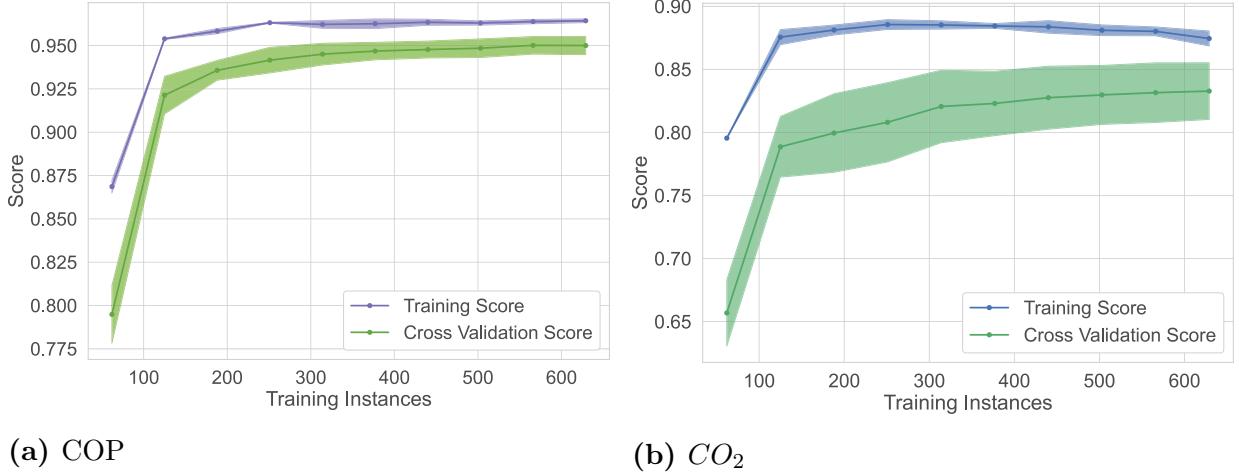


Figure 4.10: Learning curves for the final model

To summarize, this final model has very good performance, with a small bias and variance. As already noticed in 3.3.2, the CO_2 output seems difficult to predict as the performance is lower in comparison with the COP.

Figure 3.24 and Figure 3.25, Figure 3.26 and Figure 3.27 show a comparison of the other models fitted on the full predictors and XGB fitted on the reduced space. All performances of the final model are close to those of XGB on the full space.

4.1.3 Comparison with other reduced models

To further justify the choice of XGB as the final model, we implement all the other models with the relevant predictors only. Figure 4.11 and Figure 4.12 compare the training and test R^2 of the COP. Figure 4.13 and Figure 4.14 compare the training and test R^2 of the CO_2 emissions.

Note. RF is not displayed because it was not able to converge.



Figure 4.11: Reduced model comparison with violin plot of the COP R^2 on train set. 5 folds CV, 3 repetitions.

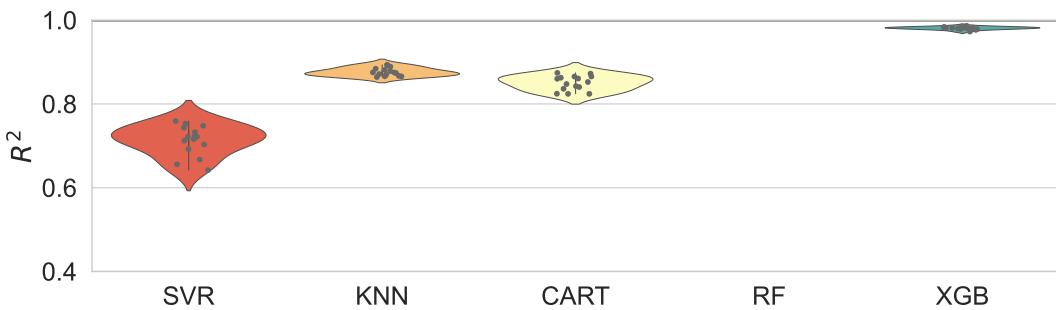


Figure 4.12: Reduced model comparison with violin plot of the COP R^2 on test set. 5 folds CV, 3 repetitions.

We observe approximately the same results as for the comparison with all predictors. Whether for the COP or the CO_2 emissions, XGB has the best performance.

4.2 Predictive space

One of the 2 objectives of this thesis is to be able to have an accurate prediction of the COP and CO_2 emissions in the least amount of time. The ML surrogate model solves this time issue: trained, it takes no time to predict the output values. Predicting the outcome for a large quantity of inputs becomes possible and can help to give a general understanding of the fluctuation of the outputs according to the predictors. To do so we span the 6-dimensional space of predictors with 8 bins per dimension within the range given in Table 3.1. We predict the outputs of these $8^6 = 2097152$ points dataset. It takes less than 7 minutes.

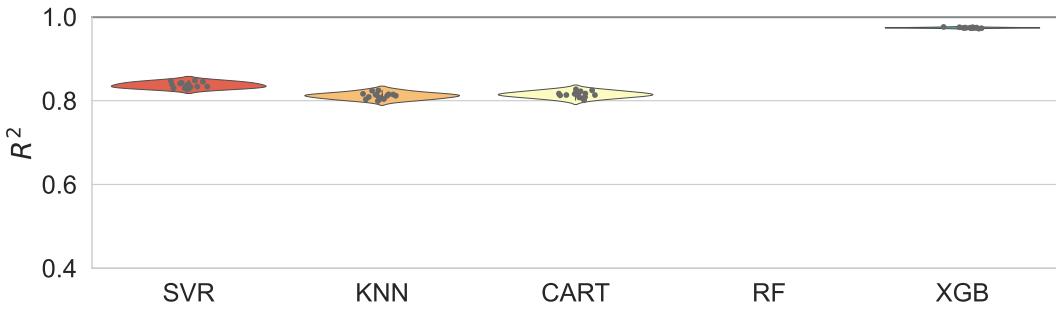


Figure 4.13: Reduced model comparison with violin plot of the $CO_2 R^2$ on train set. 5 folds CV, 3 repetitions.

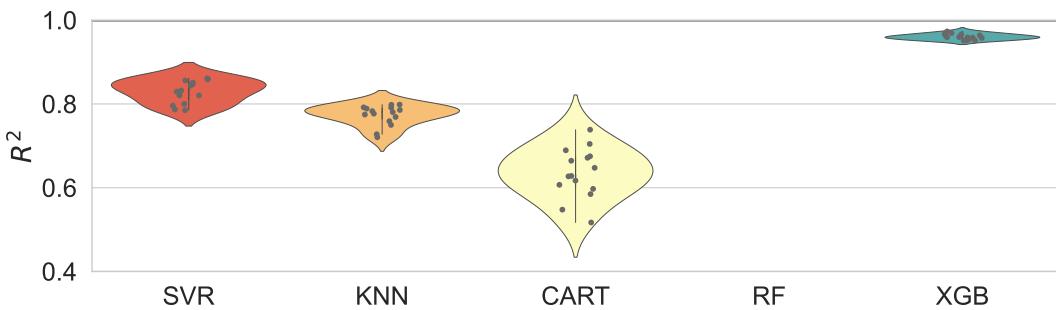


Figure 4.14: Reduced model comparison with violin plot of the $CO_2 R^2$ on test set. 5 folds CV, 3 repetitions.

Where it would take about 139810 hours (or 5825 days) with the numerical simulation (with an average running time for one simulation of 4 minutes).

Figure 4.15, Figure 4.16 and Figure 4.17 show some slices of the predicted space. In these heat-maps, we display either the COP or CO_2 in 2 dimensions of the space and fix the other 4 dimensions to the default values given in Table 3.1. The output scale is the same for every map. The lowest and the highest value of the COP and CO_2 emissions are taken as limits. In total there are 15 different heat-maps but we choose to show only a few of them.

Figure 4.15 displays the outputs according to the predictors: “SH-HP heating curve” and “DHW demand”. In this slice, when the “DHW demand” value increases, both outputs increase. Whereas when the “SH-HP heating curve” value increases, the COP decreases and the CO_2 increases.

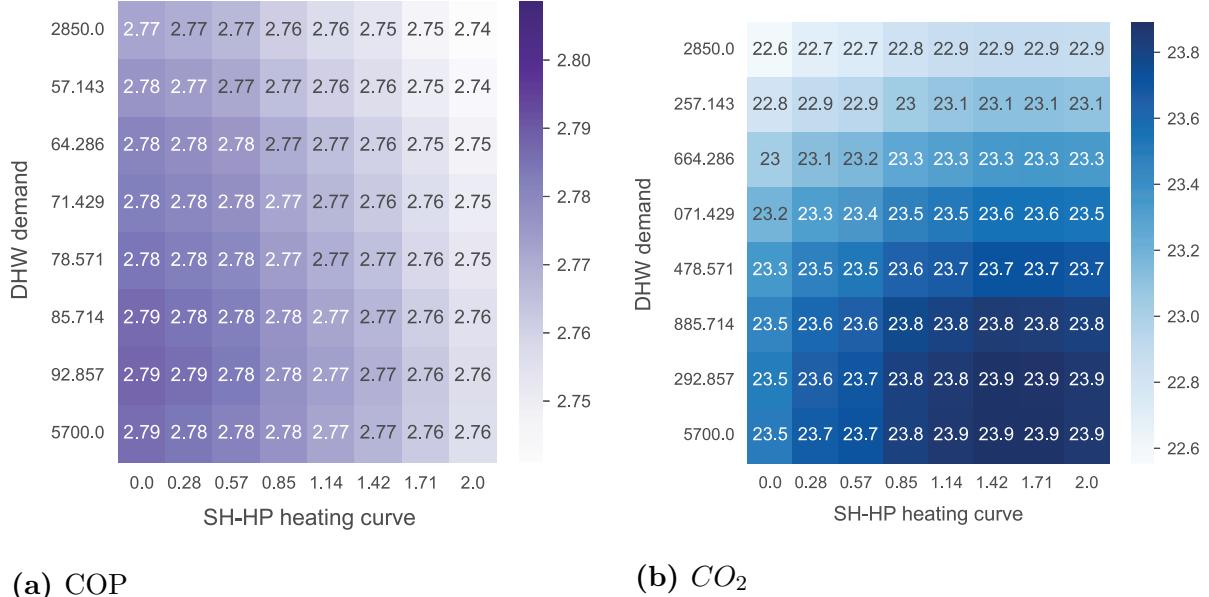
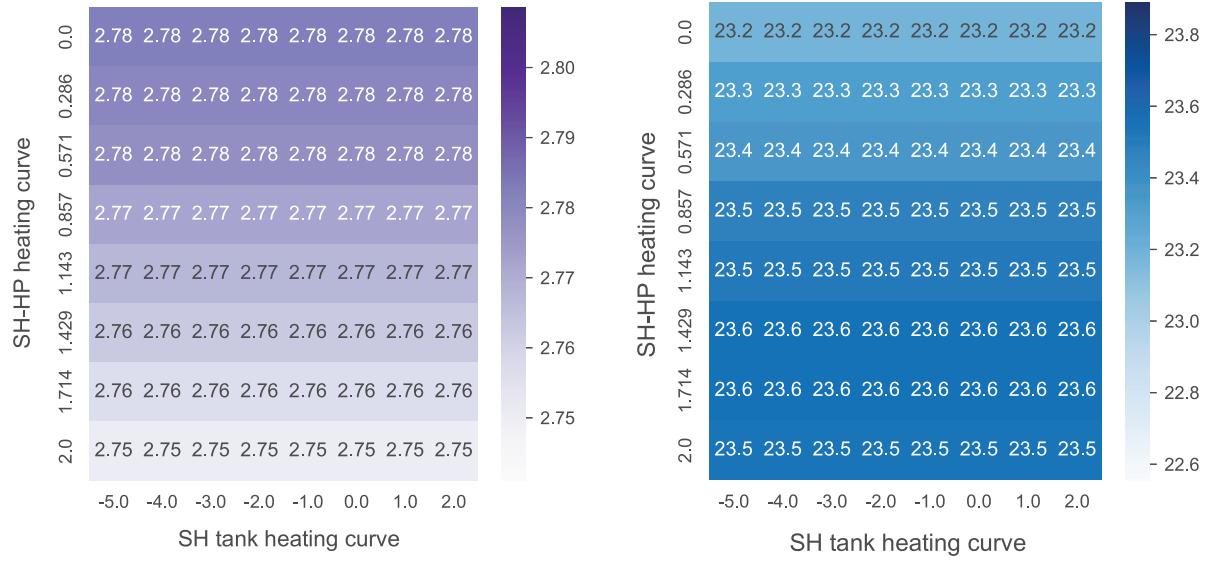


Figure 4.15: Slice of the predictive space displaying the SH-HP heating curve and DHW demand predictors

Figure 4.16 displays the outputs according to the predictors: “SH tank heating curve” and “SH-HP heating curve”. In this slice, when the “SH tank heating curve” value increases, neither output changes. Whereas when the “SH-HP heating curve” value increases, the COP decreases, and the CO_2 increases.

Figure 4.17 displays the outputs according to the predictors: “SH-HP heating curve” and “SH volume tank”. In this slice, when the “SH-HP heating curve” value increases, the COP decreases, and the CO_2 increases. Whereas when the “SH volume tank” decreases neither output changes expect a big increase for the extreme value of the COP.

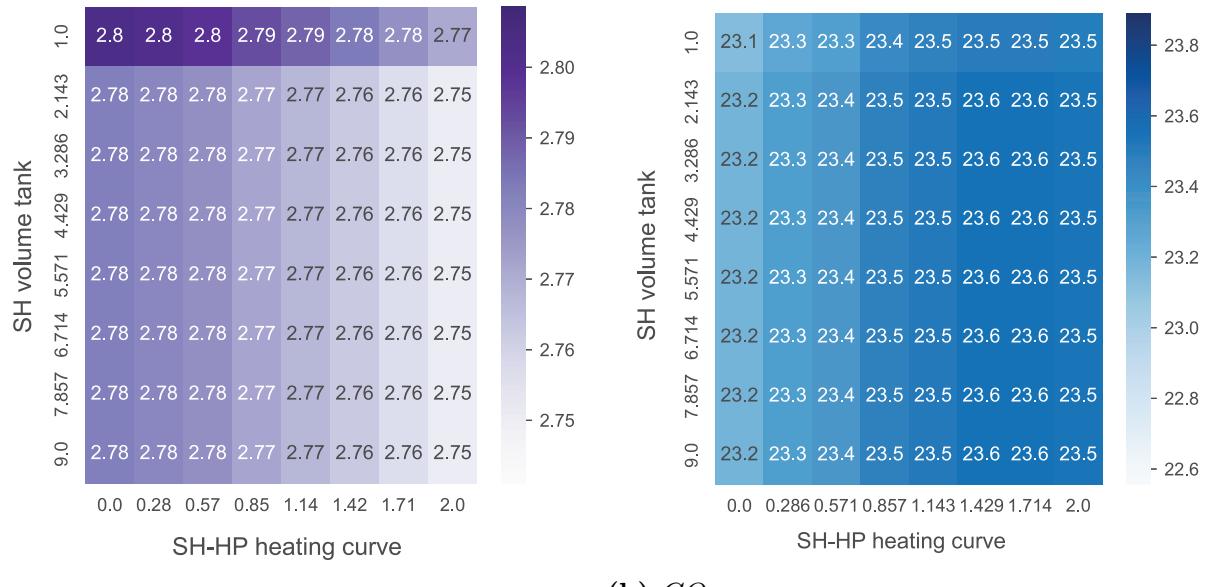
Interestingly, for some predictors, the two outputs have the same gradient and for some others, their gradient is in opposite direction. Since the optimal region is the one with high COP and low CO_2 emissions, the predictors with a gradient in opposite direction are much easier to optimize than the one with the same gradient direction. In the first case, the optimal value is at the extreme of the scale whereas for the second case a trade-off should be found. In addition to that, the gradient value can be very different across predictors.



(a) COP

(b) CO_2

Figure 4.16: Slice of the predictive space displaying the SH tank heating curve and SH-HP heating curve predictors



(a) COP

(b) CO_2

Figure 4.17: Slice of the predictive space displaying SH-HP heating curve and SH volume tank predictors

4.3 Optimization on reduced space

4.3.1 Distribution of predictors in optimal regions

We want to find a reduced range of value for the predictors that provide high COP and low CO₂ emissions. According to the heat-pump literature Calame et al. (2021), the COP is considered high if it is greater than 2.80. On the other hand, the CO₂ emission is considered reasonably low when it is lower than 23.66. Figure 4.18, Figure 4.19 and Figure 4.20 display the normalized distribution of the selected predictors (“DHW demand”, “DHW secondary flow”, “SH volume tank”, “SH-HP heating curve”, “SH tank heating curve”, “DHW set point”) that provide output of optimal regions. Default values are signaled with a black dot.

High COP

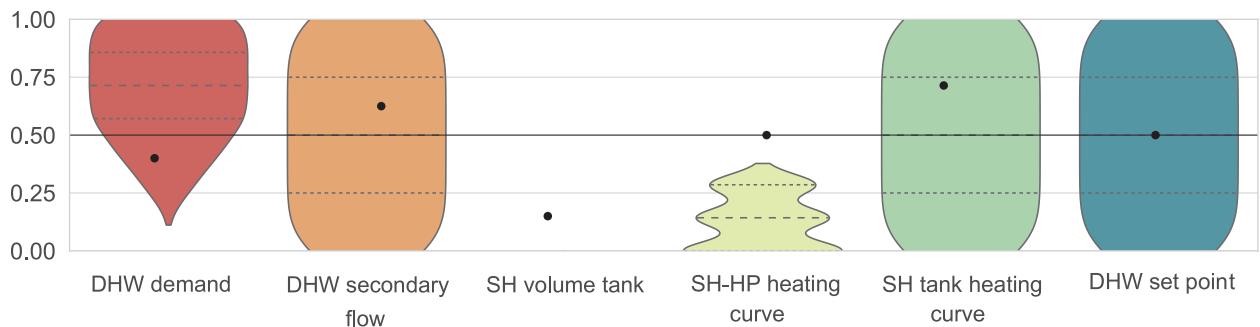


Figure 4.18: Violin plot of the normalized distribution of the predictors that provide $y_{COP} > 2.80$. Default values are signaled with a black dot.

Figure 4.18 displays the distribution of the predictors that provide a COP greater than 2.80. The predictors “DHW secondary flow”, “SH tank heating curve” and “DHW set point” are uniformly distributed across their whole range. “DHW demand” distribution is biased toward high values. “SH volume tank” and “SH tank heating curve” show a distribution very biased toward low values.

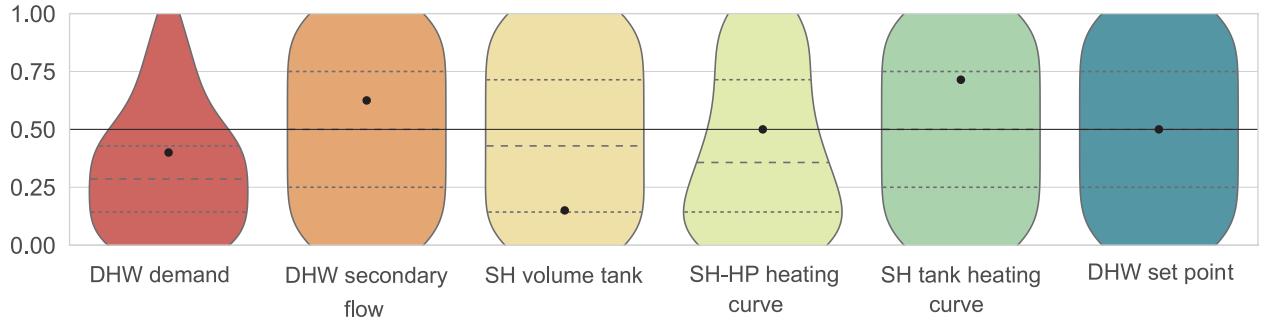


Figure 4.19: Violin plot of the normalized distribution of the predictors that provide $y_{CO_2} < 23.66$. Default values are signaled with a black dot.

Low CO_2

Figure 4.19 displays the distribution of the predictors that provide a CO_2 lower than 23.66. The predictors “DHW secondary flow”, “SH volume tank”, “SH tank heating curve” and “DHW set point” are uniformly distributed across their whole range. “DHW demand” distribution is slightly biased toward low values. “SH-HP heating curve” distribution is very slightly biased toward low values.

High COP and low CO_2

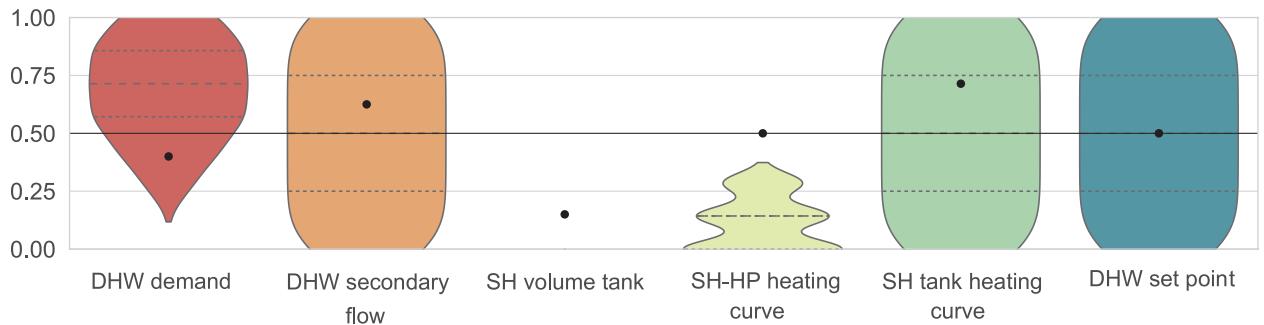


Figure 4.20: Violin plot of the normalized distribution of the predictors that provide $y_{COP} > 2.80$ and $y_{CO_2} < 23.66$. Default values are signaled with a black dot.

Figure 4.20 displays the distribution of the predictors that provide a COP greater than 2.80 and a CO_2 lower than 23.66. We find approximately the same distribution as for the high COP value.

Note: None of these distributions has a particular bias toward the default setting.

4.3.2 Results

Figure 4.20 enabled us to make the following deductions:

1. The modification of the “DHW secondary flow”, “SH tank heating curve” or “DHW set point” default value is not necessary to reach the optimized region, since their distribution in the optimized region is the same as the one of the whole space.
2. A reasonable increase of “DHW demand” and a large decrease of “SH volume tank” and “SH tank heating curve” from the setting value predict outcome in the optimal region.

The heat-maps displayed in the previous section are in accordance with these results:

- In Figure 4.15, the increase of “DHW demand” increases the COP but also the CO_2 emissions.
- In Figure 4.15, Figure 4.16 and Figure 4.17 the decrease of “SH-HP heating curve” increases the COP and decrease the CO_2 emissions.
- In Figure 4.17 the extreme decrease of “SH volume tank” increases a lot the COP.

4.3.3 Verification

In order to verify our result, we run some simulations with the inputs in the optimal region and compare the distribution of the output with the dataset.

To do so, we sample 300 times from a Gaussian distribution with the mean and standard deviation of the predictor distribution in the optimal region. Figure 4.21 shows the distribution of this dataset. Then we run the numerical simulation with this data.

Figure 4.22 shows the distribution of the outputs for data sampled in the whole space vs data sampled in the optimal space. The t-test between the COP distribution on the default space of predictor and the one on the optimal region is $5.51e - 31$. So the two distributions are different. The t-test between the CO_2 distribution on the default space of predictor and the one on the optimal region is $1.87e - 14$. So the two distributions are different.

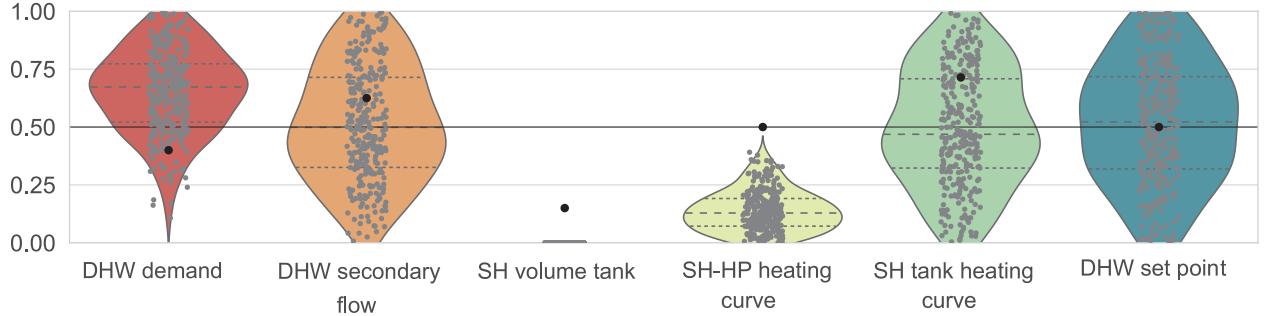


Figure 4.21: Input normalized distributions for each variable of the verification data set. Each grey dot is an input. Black dots are the normalized default values.

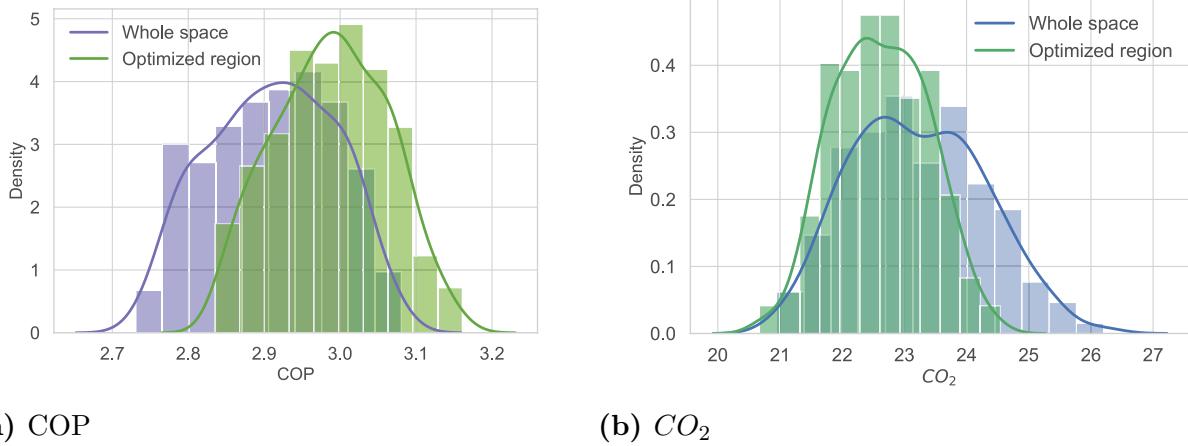


Figure 4.22: Distribution of the outputs for data sampled in the default space vs data sampled in the optimal space

Table 4.1 compare the mean and range of the two datasets for the COP and CO_2 . The mean and maximum values for the COP are larger in the optimal region in comparison to the default region. The mean and minimum values for the CO_2 are smaller in the optimal region in comparison to the default region. So, the outputs of the optimal space are clearly optimized in comparison to the default space.

Table 4.1: Comparison of the default space and optimal space for the COP and CO_2

	mean COP	max COP	mean CO_2	min CO_2
Default space	2.91	3.08	23.2	20.9
Optimal space	2.99	3.16	22.6	20.7

4.3.4 Discussion

Changeability of the predictors In practice, the values of each predictor can be more or less difficult to modify. Table 4.2 defines the predictors' different level of changeability and Figure 4.23 shows each predictor's changeability capacity.

Level	Definition
Easy	Predictor parameterized in the control system: easily changeable.
Medium	Predictor requiring system intervention but easy to execute.
Hard	Predictor necessitating a more complex intervention and a significant financial investment.
Impossible	Predictor specific to the building structure or tenant.

Table 4.2: Different levels of changeability of the predictors

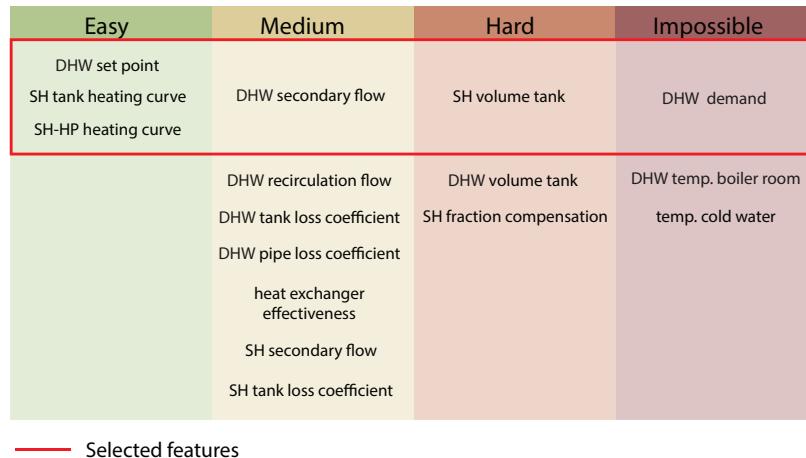


Figure 4.23: Changeability of the features

Suggestions According to Figure 4.23 the “SH-HP heating curve” is easy to change where it is hard for the “SH volume tank” and impossible for the “DHW demand”. Hence we suggest decreasing the “SH-HP heating curve” as an easy modification to the settings to archive better performance of the HP as well as lower the CO_2 emissions.

Chapter 5

Conclusion

A significant switch from fossil to renewable energy, notably through heat-pump systems, may drastically reduce building stock CO_2 emissions in tandem with declining building stock heat demand. Based on a monitoring campaign in a pilot project, numerical simulation models of the heat-pump system were built and validated. The simulation makes it possible to change specific system input parameters and track how doing so impacts the system's performance and CO_2 emissions in order to improve them. This approach is not feasible, though, because each simulation requires a lot of computer time. In this study, we suggested a surrogate machine learning model that, resembles the numerical simulation, while reducing calculation time. The objective was to identify the set of input parameters that optimizes heat-pump performance while lowering CO_2 emissions.

After analysis and comparison with six machine learning models, we selected the one with the best performance: Extreme Gradient Boosting. To run the prediction on the whole range of inputs, the predictor space was reduced. We selected the six following predictors based on feature importance analysis: **DHW demand**, **DHW secondary flow**, **SH volume tank**, **SH HP heating curve**, **SH tank heating curve**, and **DHW set point**. Predicting the outcome of a large number of inputs becomes possible and can help to give a general understanding of the fluctuation of the outputs according to the predictors. Also, we looked for a range of predictors that provide high coefficient of performance and low CO_2 emissions

(optimal input region). It came out that a reasonable increase of the “DHW demand” and a large decrease of the “SH volume tank” and the “SH tank heating curve” from the default input values predict high coefficient of performance and low CO_2 emissions. The mean and maximum values for the coefficient of performance are larger for the optimal input region in comparison to the whole input space. In the same way, the mean and minimum values for the CO_2 are smaller for the optimal input region in comparison to the whole input space.

Further directions This optimization uses only one restrictive set of data coming from the simulation. To improve its precision, one can implement a method that goes back and forth with the simulation. A good example is ActivO [Owoyele and Pal \(2021\)](#) a machine learning-based optimization algorithm for accelerating simulation-driven engine design.

Chapter 6

Appendix

6.1 Learning curves of linear model

Figure 6.1 shows the learning curves for the COP and CO_2 emissions of the linear model. The two curves converge together almost from the beginning and display a high variance.

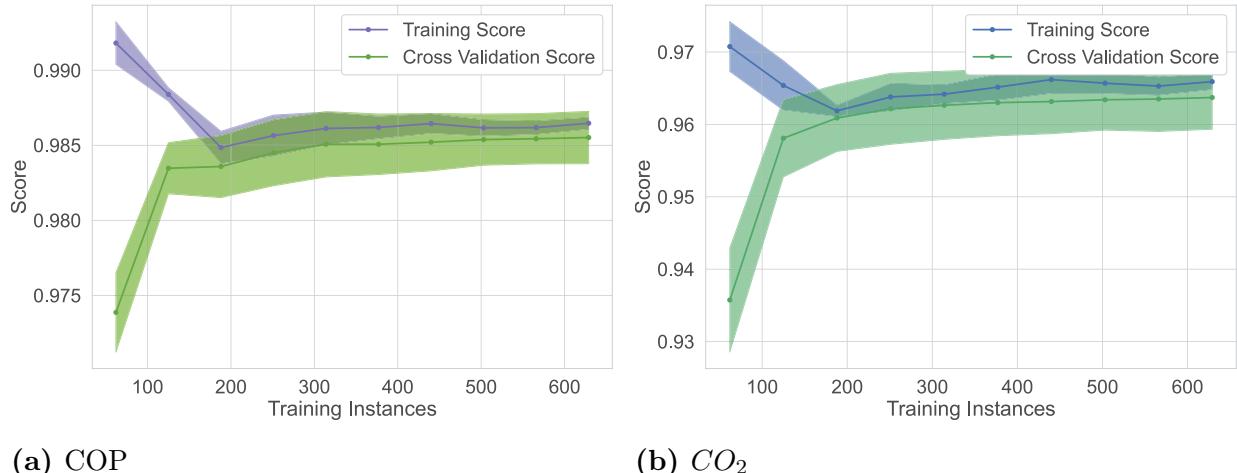


Figure 6.1: Learning curves of the linear model

6.2 Model hyperparameter selection

Figure 6.2, Figure 6.3, Figure 3.11, Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7, Figure 6.8 and Figure 6.9 show the hyperparameter selection for the COP and CO_2 emissions for SVR, KNN, CART, RF and XGB. The optimal hyperparameter value is indicated by the red line.

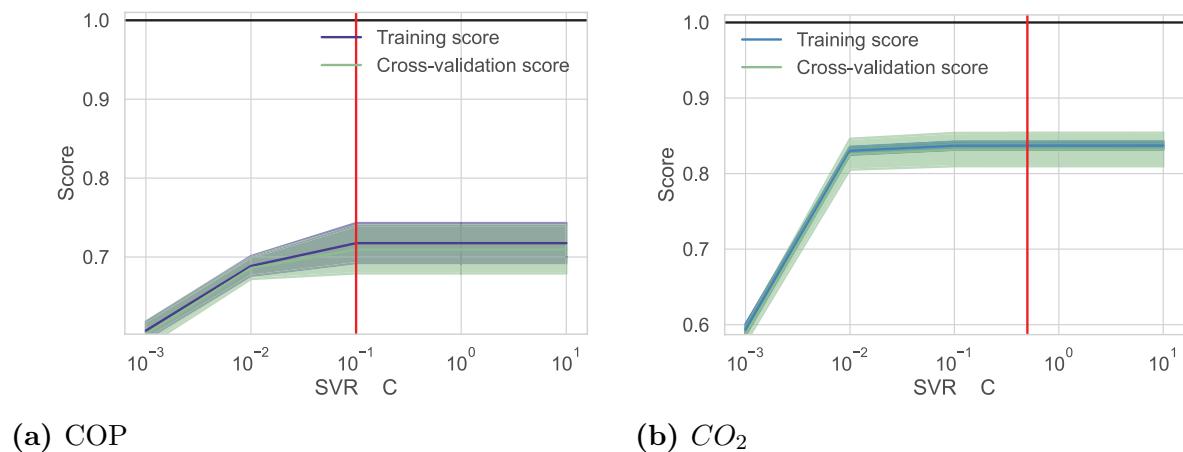


Figure 6.2: Total budget selection for SVR

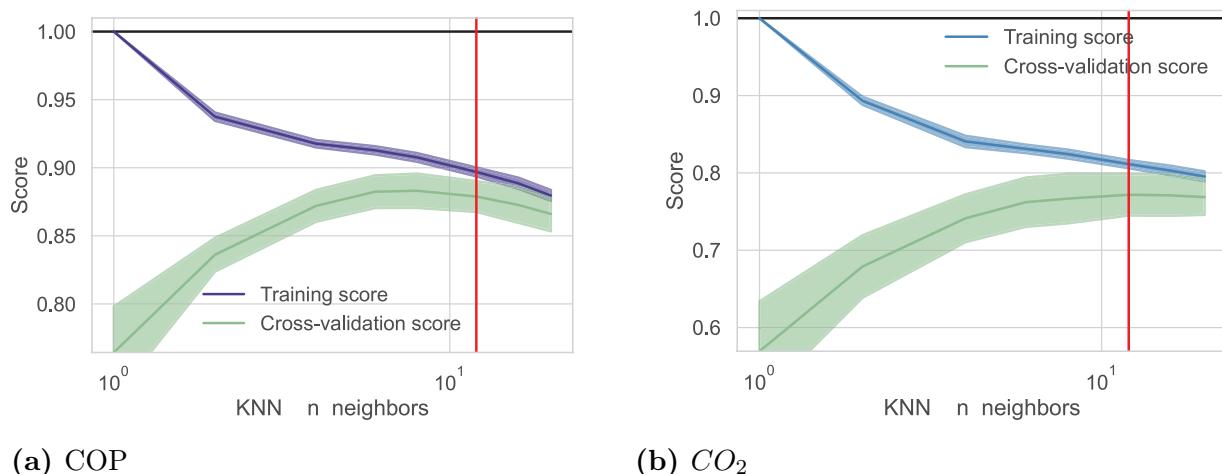


Figure 6.3: Number of neighbors selection for KNN

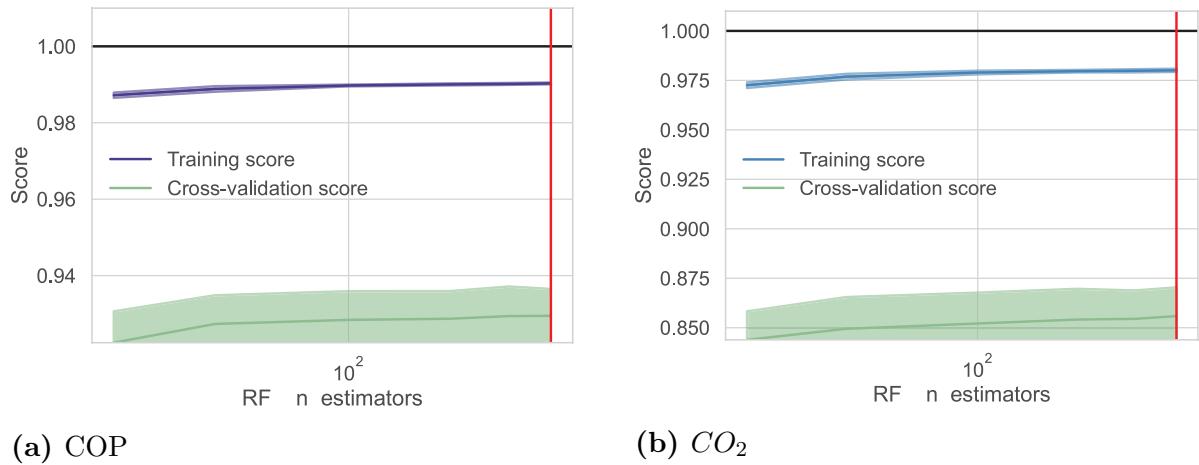


Figure 6.4: Number of estimators selection for RF

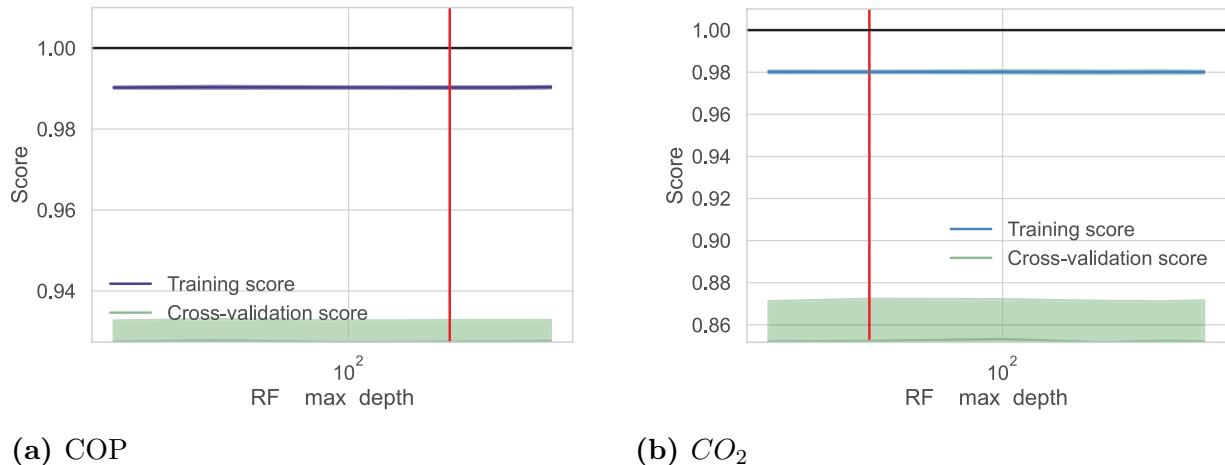
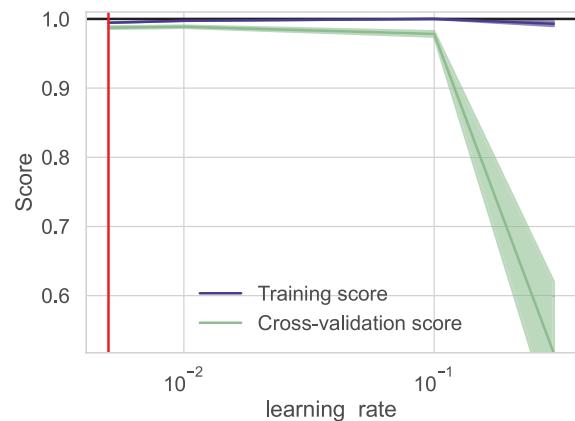


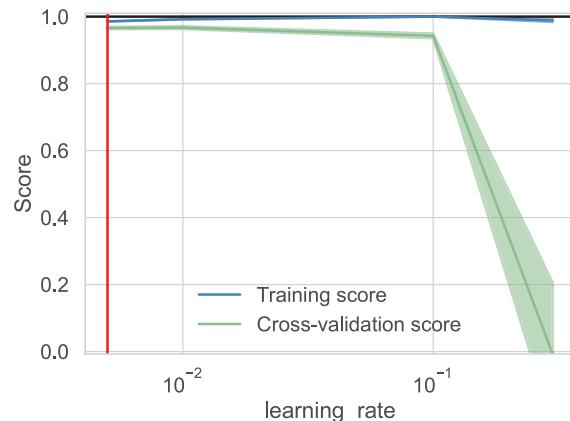
Figure 6.5: Maximum depth selection for RF

6.3 Comparison of the models with MSE

Figure 6.10 and Figure 6.11 plot the distribution of the COP MSE on the cross-validated train and test set respectively. Figure 6.12 and Figure 6.13 plot the distribution of the CO_2 MSE on the cross-validated train and test set respectively. The results are identical to those with the R^2 .

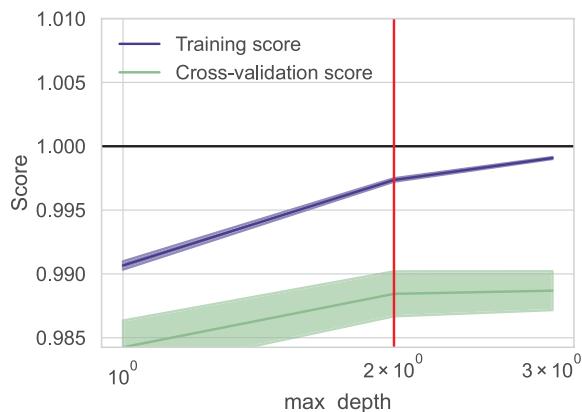


(a) COP

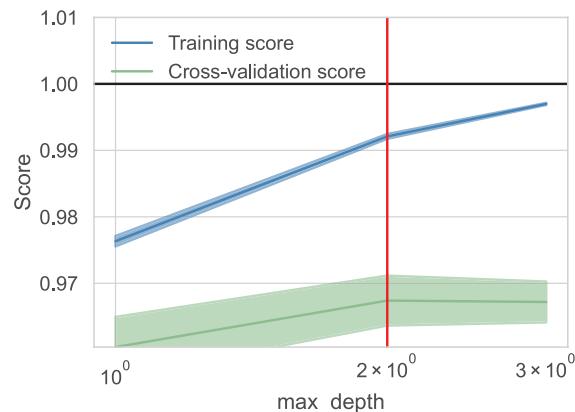


(b) CO_2

Figure 6.6: Learning rate selection for XGB



(a) COP



(b) CO_2

Figure 6.7: Maximum depth selection for XGB

6.4 Glossary

All words followed by an asterisk in the thesis are described in Table 6.1.

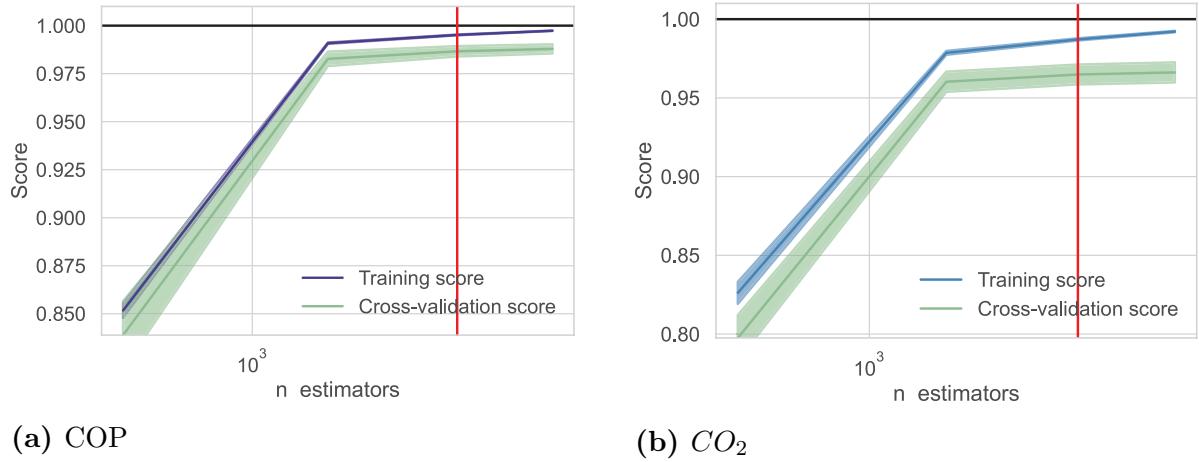


Figure 6.8: Number of estimators selection for XGB

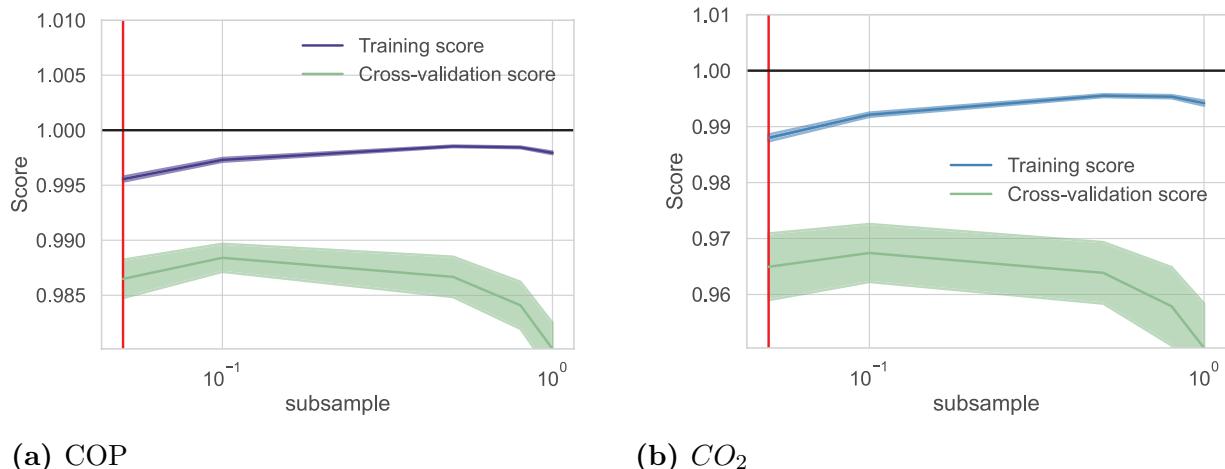


Figure 6.9: Subsample selection for XGB

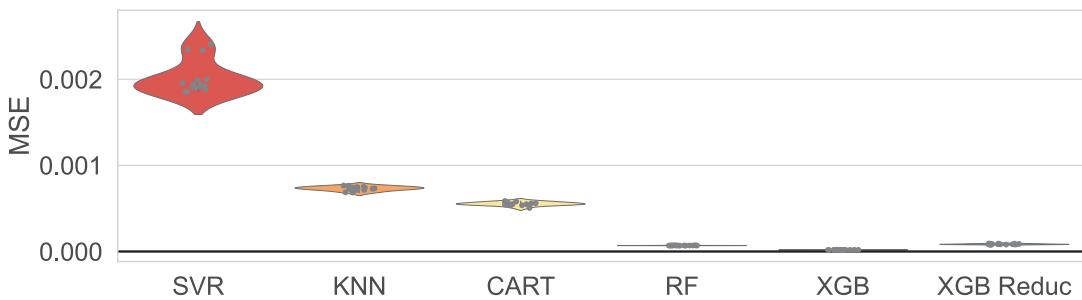


Figure 6.10: Model comparison with violin plot of the COP MSE on train set. 5 folds CV, 3 repetitions.

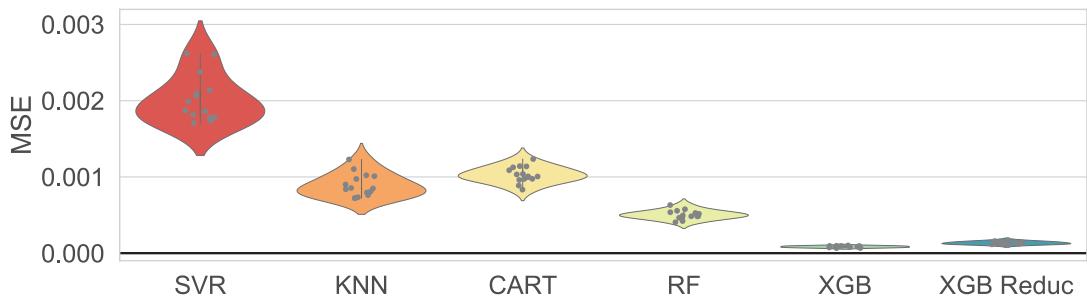


Figure 6.11: Model comparison with violin plot of the COP MSE on test set. 5 folds CV, 3 repetitions.

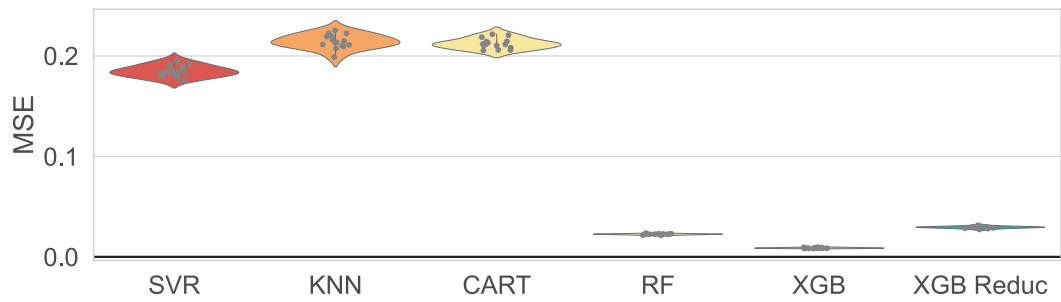


Figure 6.12: Model comparison with violin plot of the CO_2 MSE on train set. 5 folds CV, 3 repetitions.

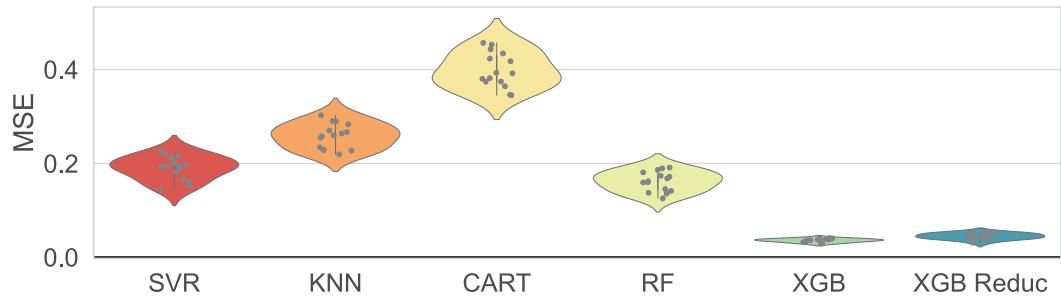


Figure 6.13: Model comparison with violin plot of the CO_2 MSE on test set. 5 folds CV, 3 repetitions.

Table 6.1: Glossary related to technical heat-pump and ML terms.

Term	Definition
COP	“The coefficient of performance calculated as the ratio of cooling capacity provided to electrical power.” Konuklu et al. (2018)
Decarbonization	“Decarbonization refers to a decrease of CO ₂ emission intensity in a trend” Sun (2005) .
Heat pump	“Heat pumps have been a well-known technology for many years and widely used for heating and cooling of residential buildings due to relatively higher energy utilization efficiencies.” Hepbasli (2018)
Latin hypercube sampling	It is a statistical technique for generating quasi-random sample distributions. It is one of the most popular sampling techniques in computer experiments, thanks to its simplicity and projective properties to high-dimensional problems: each dimensional space representing a variable is sliced into n sections, placing only one point in each section. McKay et al. (2000)
Surrogate model	“Surrogate models are used across engineering analysis to simplify complex design problems using approximations that emulate computationally demanding simulations.” Sutherland (2020)
TRNSYS	“It is an extremely flexible graphically based software environment used to simulate the behavior of transient systems. While the vast majority of simulations are focused on assessing the performance of thermal and electrical energy systems, TRNSYS can equally well be used to model other dynamic systems such as traffic flow, or biological processes.” Specialists (2019)

Bibliography

- Abdallah, M. A. A. (2014). *Optimizing the selection of sustainability measures for existing buildings*. University of Illinois at Urbana-Champaign.
- Calame, N., Cuvillier, G., Rognon, F., Montero Dominguez, O., Brischoux, P., Callegari, S. A., Hollmuller, P., Fraga, C., and Rüetschi, M. (2021). Développement et optimisation de concepts hybrides de pompes à chaleur sur l'air pour des immeubles résidentiels collectifs.
- Dallapiccola, M., Trentin, F., Barchi, G., Dipasquale, C., Fedrizzi, R., and Moser, D. (2020). Machine learning driven optimization of a hybrid electrical and thermal system. In *Proceedings of the 37th European Photovoltaic Solar Energy Conference and Exhibition, Lisboa, Portugal*, pages 7–11.
- De Sousa Fraga, C. (2017). *Heat pump systems for multifamily buildings: which resource for what demand?* PhD thesis, University of Geneva.
- Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.
- Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

- Hepbasli, A. (2018). Heat pumps. Comprehensive Energy Systems.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.
- Kalogirou, S. A. and Bojic, M. (2000). Artificial neural networks for the prediction of the energy consumption of a passive solar building. *Energy*, 25(5):479–491.
- Klein, S., Beckman, W., Mitchell, J., Duffie, J., Duffie, N., Freeman, T., Mitchell, J., Braun, J., and Evans, B. (2017). J. 436 kummer, et al, trnsys 18. a transient system simulation program; standard component library 437 overview. *Solar Energy Laboratory, University of Wisconsin-Madison: Madison, WI, USA*, 3:438.
- Konuklu, Y., Şahan, N., and Paksoy, H. (2018). 2.14 latent heat storage systems.
- Magnier, L. and Haghishat, F. (2010). Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network. *Building and Environment*, 45(3):739–746.
- McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.
- Montero, O., Brischoux, P., Callegari, S., Fraga, C., Rüetschi, M., Vionnet, E., Calame, N., Rognon, F., Patel, M., and Hollmuller, P. (2022). Large air-to-water heat pumps for fuel-boiler substitution in non-retrofitted multi-family buildings—energy performance, co₂ savings, and lessons learned in actual conditions of use. *Energies*, 15(14):5033.
- Montero, O., De Sousa Fraga, C., Callegari, S. A., and Hollmuller, P. (2020). Air-to-water heat pumps as a substitution of oil-boiler in a non-retrofitted multi-family building of the 70's. in-situ monitoring, actual energy balance and performance. In *13th IEA Heat Pump Conference 2020*.
- OFEV. (2022a). Bâtiments.

OFEV. (2022b). Inventaire des gaz à effet de serre de la suisse.

Owoyele, O. and Pal, P. (2021). A novel machine learning-based optimization algorithm (activo) for accelerating simulation-driven engine design. *Applied Energy*, 285:116455.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Seyedzadeh, S., Rahimian, F. P., Oliver, S., Rodriguez, S., and Glesk, I. (2020). Machine learning modelling for predicting non-domestic buildings energy performance: A model to support deep energy retrofit decision-making. *Applied Energy*, 279:115908.

Sharif, S. A. and Hammad, A. (2019). Developing surrogate ann for selecting near-optimal building energy renovation methods considering energy consumption, lcc and lca. *Journal of Building Engineering*, 25:100790.

Specialists, T. E. S. (2019). What is trnsys?

Sun, J. W. (2005). The decrease of co2 emission intensity is decarbonization at national and global levels. *Energy Policy*, 33(8):975–978.

Sutherland, B. R. (2020). Driving data into energy-efficient buildings. *Joule*, 4(11):2256–2258.

Thrampoulidis, E., Mavromatidis, G., Lucchi, A., and Orehounig, K. (2021). A machine learning-based surrogate model to approximate optimal building retrofit solutions. *Applied Energy*, 281:116024.

Yellowbrick (2022). Yellowbrick documentation.